VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCA

2012

Bc. Radoslav Činčala

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Sledovanie frekvencie slov na internetových spravodajských serveroch

Word Frequency in the Internet News Servers

2012 Bc. Radoslav Činčala

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Radoslav Činčala**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Sledování frekvence slov v internetových zpravodajských serverech
Word Frequency in the Internet News Servers

Zásady pro vypracování:

Tato práce se bude zabývat zpracováním článků na veřejných zpravodajských serverech, kde výstupem bude frekvence nejčastějších slov v určitém časovém úseku nebo případně na určitém zpravodajské serveru. Formát článku se na jednotlivých serverech může značně liší a strojové extrahování hlavního textu článku nemusí být jednoduché. Práce by se měla zabývat především metodami extrahování dat z článků, aby bylo možné jednoduše přidávat do sledování další zpravodajské servery.

Výsledné řešení tedy bude mít následující části:
1. Bude vytvořen robustní nástroj pro strojové extrahování dat z článků na zpravodajských serverech.
2. Vznikne nástroj, který umožní jednoduše a rychle přidat zpravodajský server do automatického sledování a strojové extrakce.
3. Extrahovaná data pak budou dále zpracována a v databázi budou uloženy frekvence jednotlivých slov spolu s dalšími souvisejícími daty tak, aby bylo možné získat statistky pro různé časové intervaly i pro různé servery.
4. Výstup extrakce dat bude možné ovlivnit seznamy stop slov a ekvivaletních slov, které bude možné jednoduše dynamicky ručně měnit.
5. Vytvoření webového uživatelského rozhraní, které umožní efektivně vyhledat frekvenci slov v daném časovém intervalu nebo na daném serveru.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radim Bača, Ph.D.**

Datum zadání: 18.11.2011
Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA * OSTRAVA * Fakulta elektrotechniky a informatiky

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave 29. jún 2012

Činčala

Rád by som na tomto mieste poďakoval pánovi Ing. Radimovi Bačovi, Ph.D., ktorý ma vďaka plnohodnotným konzultáciám viedol k správnym výsledkom a k úspešnému zvládnutiu tejto práce.

Abstrakt

Cieľom tejto práce je spracovanie článkov na verejných českých spravodajských serveroch. Výstupom je frekvencia najčastejších slov v určitom časovom intervale alebo na určitom spravodajskom serveri. Formát článkov sa na jednotlivých serveroch značne odlišuje a strojové extrahovanie hlavného textu článku nie je jednoduché. Práca sa zaoberá predovšetkým metódami extrakcie dát z článkov, aby bolo možné jednoducho pridávať do sledovania ďalšie spravodajské servery.

Výsledným riešením je vytvorenie robustného nástroja pre strojové extrahovanie dát z článkov na spravodajských serveroch a nástroj, ktorý umožňuje jednoduché a rýchle pridávanie spravodajských serverov do automatického sledovania a strojovej extrakcie. Extrahované dáta sú následne spracovávané a uložené do databázy spolu s frekvenciami jednotlivých slov a ďalšími súvisiacimi dátami tak, aby bolo možné získať štatistické údaje pre rôzne časové intervaly a pre rôzne servery.

Výstup extrakcie dát je možné ovplyvniť zoznamami stop slov a ekvivalentných slov, ktoré je možné jednoducho dynamicky meniť. Prácu s nástrojom umožňuje jednoduché webové užívateľské rozhranie, ktoré dovoľuje efektívne vyhľadávanie frekvencie slov v danom časovom intervale alebo na danom serveri.

Kľúčové slová: čas, článok, databáza, extrahovanie, frekvencia, informácia, internetová žurnalistika, java, jazyk HTML, lematizácia, rss kanál, slovo, spravodajský server, získavanie informácií.

Abstract

The aim of this work is processing of articles on public Czech news servers. Output is frequency of the most frequent words in a certain period of time or at certain news server. Format of articles is considerably different in dependence on particular server and mechanical extracting of article's main body is not easy. The work is primarily concerned with methods of extracting data from articles for purpose of easily adding of other news servers to monitoring.

The resulting solution is creation of robust tool for mechanical data extraction from articles in news servers and tool that allows easy and fast news servers adding to automatically monitoring and mechanical extraction. Extracted data are then processed and stored into a database along with the frequencies of individual words and other related data in order to obtain statistics for different time intervals and for different servers.

The output of data extraction can be influenced by lists of stop words and equivalent words, which can be easily changed dynamically. Work with tool allows simple web interface that allows efficient searching of words frequency in a given time interval or in a given server.

Keywords: time, article, database, extraction, frequency, information, internet journalism, java, HTML language, lemming, rss feed, word, news server, information retrieval.

Zoznam použitých skratiek a symbolov

API	–	Application Programming Interface
CSS	–	Cascading Style Sheets
DFD	–	Data Flow Diagram
DOM	–	Document Object Model
HTML	–	Hyper Text Markup Language
IDE	–	Integrated Development Environment
IETF	–	Internet Engineering Task Force
MIT	–	Massachusetts Institute of Technology
MS	–	MicroSoft
RDF	–	Resource Description Framework
RSS	–	Really Simple Syndication
SGML	–	Stardard Generalized Markup Language
SRDB	–	Systém Riadenia Báze Dát
SQL	–	Structured Query Language
URL	–	Uniform Resource Locator
W3C	–	World Wide Web Consortium
WWW	–	World Wide Web
XHTML	–	Extensible Hypertext Markup Language
XML	–	EXtensible Markup Language

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

20. storočie a jeho 90.roky môžme označiť ako začiatok éry *internetovej žurnalistiky*, kedy vydavatelia tradičných tlačených novín začali využívať rýchle sa vyvíjajúce a šíriace médium ako nástenku s upútavkami na svoj tlačený obsah. Reč je samozrejme o internete, s príchodom ktorého sa otvoril nový rozmer žurnalistiky. Fakt, že tradičné papierové média nemôžu svojím množstvom a rýchlosťou šírenia informácií konkurovať spravodajským webovým portálom, bol ich tvorcom ihneď zrejmý. Mnohé prieskumy ukazujú neustály pokles tvorby papierových médií a naopak pokračujúci nárast online informačných zdrojov v rámci siete internet. Inak tomu nie je ani v Českej republike [1].

Digitálne média dokážu rýchle sprostredkovať veľkému počtu ľudí nielen text a fotografie, ale na rozdiel od svojej fyzickej podoby aj zvukovú či dokonca informáciu v podobe video záznamu. Spojujú tak v sebe tlačené médium, rozhlas aj televíziu, čím sa stávajú v dnešnej dobe univerzálnym médiom pre šírenie najrozličnejších informácií z rôznych odvetví ľudskej činnosti.

Zásadnou výhodou internetového prostredia je, že spravodajský server môže poskytnúť svojím čitateľom spravodajstvo okamžite svojimi online prenosmi. Okrem tradičných prvkov spravodajstva sa na internete objavujú takisto aj celkom špecifické formy akými sú online reportáž alebo online rozhovor, ktoré v maximálnej miere umožňujú čitateľovi priamo sa zúčastniť na tvorbe správ svojimi dotazmi alebo príspevkami.

Na českej pôde sa rok po spustení komerčného využitia internetu objavuje celá skupina zaujímavých informačných zdrojov zachovávajúcich charakter novín a časopisov. Niektoré vznikli premenou svojej fyzickej podoby do digitálnej. Okrem toho v tejto dobe vznikajú aj čisto internetové periodiká. Ako prvý a najznámejší z nich bol *Neviditeľný pes Ondreja Neffa*[2], ktorý začal vychádzať 23. apríla 1996.

Internetová žurnalistika zaznamenala od svojho vzniku rapídny vývoj a pokrok s cieľom poskytnúť svojím čitateľom aktuálne informácie z diania na celom svete.
Čitateľ ma na výber z mnohých spravodajských portálov, ponúkajúcich veľké množstvo informácií deliacich sa podľa jednotlivých kategórií. Nespochybniteľnou výhodou týchto informácií je výhoda vyplývajúca z ich digitálnej podoby. To nám dáva možnosť ich strojového spracovania počítačom. Tieto informácie tak môžme zhromažďovať, triediť, vyhľadávať v nich alebo nad nimi vykonávať rozličné komplexné operácie.

Jednou z takýchto operácií je aj *sledovanie frekvencie slov v rámci českých internetových serverov*, ktorej sa venuje táto diplomová práca. Pomocou nástroja, ktorý umožní sledovanie výskytu slov, by sme mohli získať veľmi zaujímavé výsledky a vytvoriť tak novú možnosť v rámci poskytovania každodenných informácií. Plnovýznamové slová ‚majúce najvyšší počet výskytov za určité obdobie, by tak mohli podať istý obraz o dianí v spoločnosti v tom čase. Môžeme hovoriť, že by sa nám takýmto spôsobom podarilo zachytiť *‚ducha doby“*.

Na začiatku diplomovej práce (kapitola 2) sa pozrieme, čo bolo motiváciou pre jej vznik a predstavíme si niekoľko podobných projektov. V kapitole 3 sa oboznámime s technológiou RSS, ktorá zohráva veľmi dôležitú úlohu v rámci tejto práce.
Články uverejňované na českých spravodajský serveroch sú vo formáte HTML. Budeme sa ním zaoberať na začiatku kapitoly 4, kde sa takisto zamyslíme nad spôsobom, akým sa bude dať identifikovať text článku v rámci zložitej a rôznorodej štruktúry jednotlivých webových stránok.
Pri tom sa nezaobídeme bez technológie CSS predstavenej v podkapitole 4.3. Na konci kapitoly si zhrnieme nutné predpoklady pre podporu serveru a o nič menej dôležité predpoklady korektnej extrakcie textu z článku.
Následne začneme kapitolou 5 venujúcou sa špecifikácií požiadaviek nášho systému, kde pomocou jednoduchej techniky získame prehľadný zoznam funkčných požiadaviek na náš systém.
V kapitole 6 sa zameriame na analýzu a návrh systému. Popíšeme systém ako celok a postupne rozoberieme všetky jeho dôležité časti. Súčasťou tejto kapitoly bude aj analýza získaného textu článkov 6.4. Popísané budú jednotlivé časti práce s textom: tokenizácia, lematizácia a stop slová. Systém by sa nezaobišiel bez perzistentného úložiska dát. Návrh databázy vyriešime v podkapitole 6.5.
Venovať sa budeme aj samotnej implementácií, použitým technológiám a niektorým implementačným detailom v kapitole 7.
Pred samotným záverom práce poskytneme zaujímavé výsledky testov a zozbieraných štatistík (kapitola 8). V závere budú zhodnotené výsledky, prínos a možné rozšírenie celej práce.

2 Podobné projekty

V úvodnej kapitole sme načrtli problematiku, ktorou sa budeme v práci zaoberať. V tejto kapitole sa pozrieme, čím bol nápad monitorovania frekvencie slov na českých spravodajských serveroch inšpirovaný a na niektoré ďalšie podobné projekty.

2.1 Google Trends a Google Insights for Search

Jednými z najznámejších projektov podobného zamerania sú služby *Google Trends* [3] a *Google Insights for Search* [4] , ktorými bola inšpirovaná táto práca. Z názvu vyplýva, že ide o produkty z dielne prevádzkovateľa najväčšieho svetového internetového vyhľadávača, s ktorým tiež úzko súvisia. Jedná sa o dva veľmi podobné nástroje, ktoré pracujú nad rovnakými dátami, pričom Google Insights for Search na rozdiel od Google Trends ponúka viacero pokročilých rozšírení , väčšiu flexibilitu a funkcionalitu. Je zameraný na špecifickejšiu skupinu užívateľov zaoberajúcich sa výskumom alebo reklamou v internetovom prostredí.

Tieto nástroje slúžia k analýze a zrovnávaniu počtu výskytov vyhľadávaných výrazov a umožňujú grafické zobrazenie popularity kľúčových slov podľa frekvencie ich vyhľadávania vo vyhľadávači Google. Kľúčových slov môže byť aj viacero, čo nám umožňuje vzájomne ich porovnávať. Vstupné dáta je možné obmedziť napr. ohraničením na určité časové obdobie alebo región. Geografické obmedzenie tak umožňuje zobraziť napr. štatistiky vyhľadávania iba pre ČR.

Štatistiky vyhľadávania zobrazujú výsledky iba pre vyhľadávacie dotazy, ktoré dosiahli určitej hranice výskytu. Dáta sú aktualizované denne. K výrazným špičkám grafov ponúkajú pre hojne vyhľadávané pojmy aj významné udalosti, ktoré vznik extrémov spôsobili, napr. oznámenie nového produktu firmy a podobne. Cieľom tohto nástroja je sledovať trendy vo vyhľadávaní. Štatistiky vyhľadávania od Googlu je možné použiť pre výber marketingových oznamov, skúmanie sezónnych vplyvov, vytváranie asociácie so značkou atď.

Podobnosť s našou prácou je v monitorovaní a získavaní štatistík pre jednotlivé slová. Odlišnosť spočíva najmä v zdroji, z ktorého sú jednotlivé slova získavané. Kým služby od Googlu sa zaoberajú spracovaním kľúčových slov, ktoré užívatelia zadávajú priamo do vyhľadávača, v tejto práci budeme narábať so slovami, ktoré tvoria obsahy článkov na českých spravodajských serveroch.
Z toho vyplýva, že slová bude potrebné z článku najskôr získať, aby s nimi bolo možné ďalej pracovať. Výstup extrakcie bude možné ovplyvniť zoznamami stop slov a ekvivalentných slov, ktoré budeme mať možnosť jednoducho dynamicky meniť a prispôsobovať tak naším potrebám. Čo presne sa skrýva za týmito špecifickými skupinami slov sa dozvieme v neskoršom štádiu práce. Podobne ako služby od Googlu budeme poskytovať okrem rebríčka najčastejších slov aj grafické znázornenie ich frekvencie na časovej osi. Rebríček najčastejších slov bude možné špecifikovať podľa rôznych kritérií ako napr. čas, kategória alebo konkrétny server.

2.2 Wiktionary - Frekvenčné zoznamy

„Wiktionary, the free dictionary“ je projekt založený na spolupráci a spoločnom udržiavaní, ktorý si kladie za cieľ poskytovať voľne prístupný slovník v každom jazyku s definíciami, etymológiou a výslovnosťou. Obsah Wikislovníka je chránený *„GNU Free Documentation Licence“*, čo znamená, že je slobodný a zadarmo. Mnohojazyčný slovník vo svojej anglickej verzii obsahujúci anglické definície takmer 3 miliónov slov z vyše 450 jazykov.
Súčasťou tohto slovníku je aj zoznam frekvencie slov z rôznych doménových oblastí pre rôzne jazyky[5]. Uvedené frekvenčné zoznamy sčítavajú jednoznačné pravopisné slová vrátane ekvivalentných foriem. Napr. slovo „byť“ je reprezentované tvarmi „je“, „sú“, „boli“, atď. Z toho vyplýva, že ekvivalentnými formami ohybného slova českého jazyka rozumieme všetky jeho ohýbané tvary (skloňovanie, časovanie, atď.).

Frekvencia slov Českého národného korpusu

Pre češtinu je tu uvedená frekvencia slov Českého národného korpusu. *Korpus je súbor počítačovo uložených textov, ktorý slúži k jazykovému výskumu.* Český národný korpus je akademický projekt zameraný na budovanie rozsiahleho počítačového korpusu predovšetkých písanej češtiny. Pracuje na ňom Ústav Českého národného korpusu na Filozofickej fakulte Univerzity Karlovej v Prahe. Jedná sa o zrovnanie frekvenčných zoznamov z korpusu SYN2000, SYN2005 a SYN2010[6]. Sú to synchrónne reprezentatívne korpusy súčasnej písanej češtiny. Každý z nich obsahuje 100 miliónov textových slov („tokenov“). Tieto korpusy však majú okrem spomenutých zhodných rysov tiež množstvo rysov rozdielnych, ktoré sa týkajú ako zloženia textov, tak aj ich spracovania. Jednotlivé slová sú lemované a výsledkom je zoznam najčastejšie sa vyskytujúcich slov.
Podobnosť s naším projektom je viac-menej jasná, jedná sa však opäť o iný dátový zdroj. Okrem toho získanie frekvenčného zoznamu slov Českého národného korpusu je jednorazová záležitosť, kým náš nástroj bude spúšťaný v pravidelných časových intervaloch, aby analyzoval nové články a získaval z nich nové dáta.

Ako pre češtinu, tak aj pre slovenčinu, sú tu ďalej uvedené zoznamy najčastejších slov používaných vo filmových titulkoch pochádzajúcich zo serveru „Opensubtitles.org“ [7].

Ukázali sme si niekoľko tematicky podobných projektov vzhľadom k tejto práci. V nasledujúcich kapitolách sa začneme venovať samotnej problematike a procesu riešenia práce.

3 RSS ako zdroj článkov

Pred tým, ako sme sa začali zaoberať strojovým spracovaním samotných článkov, bolo potrebné vyriešiť jednu z najdôležitejších otázok: *Čo bude zdrojom článkov pre daný server?*

Bol vykonaný rozsiahly prieskum spravodajských serverov s cieľom nájsť vhodný, jednoduchý, ľahko prístupný a univerzálny zdroj resp. strojovo čitateľné rozhranie článkov pre všetky servery. Do úvahy pripadalo spravodajstvo e-mailom, ktoré však nie je poskytované všetkými spravodajskými servermi. Z toho dôvodu bola takáto možnosť vylúčená. Ako už názov kapitoly napovedá, zdrojom článkov sa po konečných rozvahách stali *RSS informačné kanály*.
Pretože RSS je jedným zo základných stavebných kameňov celej práce, venujeme mu v tejto kapitole dostatočnú pozornosť. Povieme si, čo je to RSS kanál a aké výhody prináša. Ozrejmíme jeho štruktúru a nakoniec dáme RSS do súvislosti so spravodajskými servermi, aby sme zistili ako je RSS poskytované používateľom.

3.1 Really Simple Syndication

RSS patrí do rodiny XML formátov. Je určených pre čítanie noviniek v rámci webových stránok, všeobecnejšie pre syndikáciu (združovanie) obsahu. Táto technológia umožňuje užívateľom internetu prihlásiť sa k odberu noviniek z webu, ktorý ponúka RSS zdroj označovaný tiež ako *„RSS feed“* alebo *„RSS kanál“*.
RSS zdroj sa väčšinou vyskytuje na stránkach, na ktorých sa obsah dynamicky v čase mení a pridáva veľmi často. Typickým príkladom sú spravodajské servery.
Autori stránok na internete uverejnia výstup stránky vo formáte XML, ktorý potom môžme pomocou špeciálneho programu v pravidelných intervaloch kontrolovať a následne užívateľa informovať, že bol pridaný nový obsah na webovej stránke. Takýto program, nazvime ho „RSS čítačka“, bol vytvorený aj v rámci nášho softwarového nástroja, aby sme boli schopní monitorovať stávajúce resp. nové články na spravodajských serveroch.

Cieľom RSS formátu je jednoduchosť a ľahká pochopiteľnosť. Prvá verzia s označením RSS 0.90 bola vyvinutá v roku 1999 spoločnosťou Netscape, ktorá je všeobecne známa svojím webovým prehliadačom Netscape Navigator. Akronym RSS vtedy znamenal *„RDF Site Summary“*, pretože táto prvotná verzia bola postavená nad modelom metadát RDF.

Momentálne najpoužívanejšej verzie RSS 2.0 sme sa dočkali v roku 2002. Vznikom tejto verzie bol zmenený význam RSS na *„Really Simple Syndication“*. V súčasnosti je špecifikácia RSS 2.0 [8] ponúkaná pod licenciou *„Crative Commons“*. **Ďalej sa budeme zaoberať len s RSS verzie 2.0 v spojitosti so spravodajskými servermi a budeme vychádzať z [9].**

```
<?xml version='1.0' encoding='UTF-8'?>
<?xml-stylesheet type='text/xsl' href=' http: // idnes.cz.feedsportal.com/xsl/eng/rss.xsl '?>

<rss xmlns:itunes="http: // www.itunes.com/dtds/podcast-1.0.dtd" xmlns:dc="http://purl.org/dc/
    elements/1.1/" xmlns:taxo="http: // purl.org/rss/1.0/ modules/taxonomy/" xmlns:rdf="http://www.
    w3.org/1999/02/22-rdf-syntax-ns#" version="2.0">
  <channel>
    < title >Zprávy iDNES.cz - Přehled nejnovějších událostí z domova i ze světa</title>
    <link>http: // zpravy.idnes.cz/</link>
    <description>Nejrychlejší zpravodajství na českém internetu, události z domova i celého
        světa</description>
    <language>cs</language>
    <copyright>© Copyright MAFRA a.s. 1998 - 2012</copyright>
    <pubDate>Sat, 21 Apr 2012 13:14:26 GMT</pubDate>
    <lastBuildDate>Sat, 21 Apr 2012 13:14:26 GMT</lastBuildDate>
    <ttl>2</ttl>
    <image>
      < title >Zprávy iDNES.cz - Přehled nejnovějších událostí z domova i ze světa</title>
      <url>http: // gidnes.cz/u/loga-n4/idnes.gif</url>
      <link>http: // zpravy.idnes.cz/</link>
    </image>

    <item>
      < title >Obama rozjímal v autobuse, kde Parksová odmítla pustit sednout bělocha</title>
      <link>http: // idnes.cz.feedsportal.com/c/34387/f/625936/s/1e991c2f/l/...</link>
      <description>První afroamerický prezident USA se na chvíli pokusil vcítit se ... </
          description>
      <category domain="http://zpravy.idnes.cz/zahranicni.aspx">Zprávy - Zahraniční</
          category>
      <pubDate>Sat, 21 Apr 2012 13:01:00 GMT</pubDate>
      <comments>http://zpravy.idnes.cz/diskuse.aspx?iddiskuse=
          A120421_143630_zahranicni_ipl</comments>
      <guid isPermaLink="false">A120421_143630_zahranicni_ipl</guid>
    </item>
    ...
  </channel>
</rss>
```

Výpis 1: Ukážka RSS kanálu

V ukážkovom XML dokumente vo formáte RSS 2.0 pochádzajúcom zo serveru IDnes.cz, vidíme jeho štruktúru. Začiatok dokumentu je vyhradený pre informácie o verzii XML. Na najvyššej úrovni je celý dokument uzatvorený do práve jedného elementu `<rss>`, do ktorého je vnorený element `<channel>`. Nasledujú informácie o kanále.

Nebudeme sa zaoberať vyčerpávajúcim výpisom informácií o všetkých možných podelementoch elementu `<channel>`. Popíšeme si len tie, ktoré figurujú v ukážkovom príklade, aby bol jasný ich význam.
Prehľad týchto párových elementov môžme vidieť v tabuľke 1. Oveľa dôležitejšie sú jednotlivé články, ktoré sú v RSS kanále prezentované pomocou párovej značky `<item>`. Sú súčasťou svojho rodičovského elementu `<channel>`. Ukážkový príklad obsahuje jeden

Element	Popis
`<title>`	Pomenovanie RSS kanálu v podobe čitateľnej pre človeka.
`<link>`	URL serveru.
`<description>`	Popis RSS kanálu.
`<language>`	Skratka jazyka, v ktorom je kanál napísaný (`cs`) pre češtinu. Zoznam používaných skratiek nie je v súlade s medzinárodne platnými normami *ISO Languages*(ISO 639-2T [10]).
`<copyright>`	Informácia o autorských právach.
`<pubDate>`	Dátum publikovania obsahu RSS kanálu.
`<lastBuildDate>`	Čas poslednej aktualizácie obsahu.
`<ttl>`	Číslo udávajúce počet minút. Určuje, ako dlho môže byť kanál uložený vo vyrovnávacej pamäti pred novou aktualizáciou zo zdroja.
`<image>`	Informácie o obrázku, ktorý môže kanál zastupovať.

Tabulka 1: Obsah elementu `<channel>` v RSS (vnorené elementy)

Obrázok 1: Značka pre RSS

článok s názvom: *„Obama rozjímal v autobuse, kde Parksová odmítla pustit sednout bělocha“* . Články sú špecifikované ďalšími elementmi, pričom všetky sú nepovinné, avšak musí byť prítomný aspoň jeden z dvojice `<title>` alebo `<description>`. Tie najdôležitejšie, ktoré nás budú najviac zaujímať, sú obsahom tabuľky 2. V popise k jednotlivým elementom je stručne uvedené aj to, ako využijeme obsah elementu.

3.2 RSS a spravodajské servery

Samozrejmosťou v dnešnej dobe je, že každý spravodajský server ponúka svoje RSS kanály, ako možnosť sledovania najaktuálnejších článkov. Odkaz na ne nájdeme buď to na začiatku, alebo oveľa častejšie na konci stránky. RSS kanály sú neodmysliteľné spojené so svojou značkou (obr. 1), ktorá nás informuje o prítomnosti RSS na danom servery a takisto slúži ako odkaz pre prístup k nim.

Určite je zaujímavé pozrieť sa aj na formu, akou sú RSS kanály poskytované. Používané sú dve možnosti. Prvou z nich je možnosť, kedy po zvolení odkazu na RSS dostaneme kompletný zoznam všetkých RSS kanálov, rozdelených do jednotlivých kategórií, ako

Element	Popis
`<title>`	Titulok článku. *Použijeme ho ako zdroj pre titulok článku, ktorý nebudeme získavať z webovej stránky ale z RSS kanálu.*
`<link>`	URL adresa článku. *Využijeme ju na prístup k článkom.*
`<description>`	Stručný prehľad o článku (prvý odsek). *Využijeme ho k identifikácií prvého odseku článku na webovej stránke.*
`<pubDate>`	Čas zverejnenia článku. *Bude zviazaný ako časový údaj s článkami a jednotlivými slovami článku.*
`<guid>`	Reťazec jednoznačne identifikujúci článok. Nie je určený presný formát tohto elementu. Vydavatelia si volia vlastný tvar. Ak je hodnota voliteľného atribútu `isPermaLink` nastavená na hodnotu `true` alebo nie je uvedená, čítačka môže predpokladať, že hodnota elementu je trvalý odkaz na daný článok. V opačnom prípade nastavujeme hodnotu atribútu na `false`. *V našej RSS čítačke použije hodnotu tohto elementu pri rozhodovaní, či sa jedná o celkom nový ešte nespracovaný článok.*

Tabulka 2: Dôležité elementy popisujúce články v RSS

môžeme vidieť na ukážkovom obrázku 2. Jedná sa o ukážku zoznamu RSS kanálov zo serveru „IDnes.cz“. Zoznam obsahuje mimo viditeľných aj ďalšie kategórie a kanály. Každý kanál ma priamo uvedenú svoju URL adresu, ktorú budeme používať ako vstup pre RSS čítačku.

Druhou používanou možnosťou je, že po zvolení odkazu na RSS sa dostaneme priamo na XML súbor RSS kanálu. Takto to funguje napr. na servery „tn.nova.cz“. Ak chceme získať RSS kanál pre kategóriu „Šport“, musíme v menu webovej stránky zvoliť možnosť „Šport“ a následne hľadať odkaz na RSS. Tým sa dostaneme na požadovaný kanál. Neexistuje žiaden zoznam kanálov, preto ich získavame jednotlivo z informačných kategórií serveru.
Prvá z používaných možností je z používateľského hľadiska oveľa prívetivejšia, pretože máme k dispozícií celý zoznam RSS kanálov a nemusíme sa namáhať ich hľadaním.

Čo sa týka kategórií RSS kanálov, každý server ponúka *jeden všeobecný RSS kanál*, tj. kanál do ktorého prichádzajú články zo všetkých spravodajských kategórií. Prvotne sa kvôli zjednodušeniu pracovalo s týmto kanálom. Okrem univerzálneho RSS kanálu, servery ponúkajú aj kanály rozdelené do jednotlivých kategórií. V neskoršej fáze sa preto začalo pracovať s týmito špecifickými kategóriami RSS kanálov. To nám v konečnom dôsledku

Přehled nabízených RSS zdrojů

ZPRAVODAJSTVÍ iDNES.cz

Zprávy iDNES.cz	http://servis.idnes.cz/rss.aspx?c=zpravodaj
Sport iDNES.cz	http://servis.idnes.cz/rss.aspx?c=sport
Fotbal iDNES.cz	http://servis.idnes.cz/rss.aspx?c=fotbalh
Hokej iDNES.cz	http://servis.idnes.cz/rss.aspx?c=hokejh
Tenis iDNES.cz	http://servis.idnes.cz/rss.aspx?r=tenis
Volejbal iDNES.cz	http://servis.idnes.cz/rss.aspx?r=volejbal
Basket iDNES.cz	http://servis.idnes.cz/rss.aspx?c=basket
Ekonomika iDNES.cz	http://servis.idnes.cz/rss.aspx?c=ekonomikah
Kultura iDNES.cz	http://servis.idnes.cz/rss.aspx?c=kultura
Kavárna iDNES.cz	http://servis.idnes.cz/rss.aspx?r=kavarna

REGIONÁLNÍ ZPRAVODAJSTVÍ iDNES.cz

Praha a střední Čechy	http://servis.idnes.cz/rss.aspx?c=prahah
Brno a jižní Morava	http://servis.idnes.cz/rss.aspx?c=brnoh
České Budějovice a jižní Čechy	http://servis.idnes.cz/rss.aspx?c=budejovice
Královehradecký kraj	http://servis.idnes.cz/rss.aspx?c=hradec
Jihlava a Vysočina	http://servis.idnes.cz/rss.aspx?c=jihlava
Karlovy Vary a Karlovarský kraj	http://servis.idnes.cz/rss.aspx?c=vary
Liberecký kraj	http://servis.idnes.cz/rss.aspx?c=liberec
Olomoucký kraj	http://servis.idnes.cz/rss.aspx?c=olomouc
Ostrava a Moravskoslezský kraj	http://servis.idnes.cz/rss.aspx?c=ostrava
Pardubice a Pardubický kraj	http://servis.idnes.cz/rss.aspx?c=pardubice
Plzeňský kraj	http://servis.idnes.cz/rss.aspx?c=plzen
Ústí nad Labem a Ústecký kraj	http://servis.idnes.cz/rss.aspx?c=usti
Zlínský kraj	http://servis.idnes.cz/rss.aspx?c=zlin

Obrázok 2: IDnes.cz - RSS kanály

umožnilo triediť a vyhľadávať zozbierané štatistiky podľa ďalšieho dôležitého kľúča - *podľa kategórie*.

Zaujímať nás môže aj množstvo článkov obsiahnutých v kanále. RSS kanál môže obsahovať ľubovoľné množstvo článkov. Avšak typickým množstvom článkov prítomných v RSS kanále je medzi 20 a 30. Výnimočne sa vyskytujú ako malé počty (približne 5 článkov) tak naopak vysoké počty článkov (približne 100).

Články sú v RSS zoradené do fronty s permanentnou veľkosťou. S príchodom nového článku do kanálu z neho vypadne najstarší článok. *Ak sa jedná o článok zachytávajúci nejakú dôležitú udalosť, môže byť po jeho vypadnutí do RSS kanálu znovu vložený ako nový článok z dôvodu jeho dôležitosti a opätovnej propagácie.* Na to bude potrebné myslieť pri práci s kanálom, aby sme boli schopní takýto článok rozpoznať a vylúčiť ho z nadbytočného, redundantného spracovania. Na tento účel využijeme element `<guid>` obsahujúci reťazec jednoznačne identifikujúci každý článok.

Fakt, že dôležité články prichádzajú do kanálu viac krát, umožňuje získať zoznam najvýznamnejších článkov, ktoré odrážajú to najdôležitejšie dianie v spoločnosti. To bolo využité pri tvorbe programu, čím bola pridaná ďalšia veľmi zaujímavá funkcionalita systému. Takto budeme schopní vyhľadávať články so špecifickými požiadavkami, ako je napr. kategória a konkrétny deň. Docielilo sa tak veľmi plnohodnotné využitie systému pre praktické účely.

Ostáva spomenúť ešte jedno veľmi dôležité pozorovanie, ktoré je v rámci RSS kanálov spravodajských serverov potrebné uviesť. Reč je o skutočnosti, že jeden článok môže byť publikovaný naraz vo viacerých kanáloch. Predstavme si článok o dôležitom futbalovom zápase. Tento článok bude samozrejme publikovaný v RSS kanále pre kategóriu futbal, ale zároveň v RSS kanále pre kategóriu šport. Môže nastať dokonca aj taký prípad, kedy bude článok navyše publikovaný aj v kanále z regionálnej kategórie podľa toho, kde sa zápas odohrával. Našou úlohou bude takéto články rozpoznať a vyhnúť sa tak nadbytočnému spracovaniu.
Okrem toho to bude mať aj veľmi dôležitý vplyv na poradie pridávaných kanálov do spracovania. Ak budeme chcieť, aby článok patril do kategórie futbal a zároveň šport, musíme najskôr pridať RSS kanál pre kategóriu futbal označený takisto kategóriou šport a až následne kanál pre kategóriu šport (jeden RSS kanál môže patriť do viacerých kategórií). Článok tak bude spracovaný v rámci kategórie futbal a šport. V rámci samostatnej kategórie šport už nebude opätovne spracovávaný.

Táto kapitola poskytla dostatočné informácie pre pochopenie funkčnosti RSS aj vo vzťahu k spravodajským serverom. Teóriu využijeme pri návrhu a tvorbe RSS čítačky.

4 Html formát a spravodajské servery

Pretože spravodajské servery, s ktorými sa pracovalo, používajú rôzne verzie jazyka HTML, je dôležité si povedať niečo o tomto jazyku, jeho verziách a rozdielmi medzi nimi, aby sme sa mohli následne venovať problematike automatického strojového spracovania článkov na jednotlivých spravodajských serveroch. Pozrime sa preto stručne na vývoj jazyka HTML.

4.1 Vývoj Jazyka HTML

Hypertextový značkovací jazyk HTML je hlavným jazykom učeným na vytváranie webových stránok v rámci systému WWW. Umožňuje publikovať na internete dokumenty, ktoré sú zobraziteľné vo webových prehliadačoch. Vývoj HTML je s vývojom prehliadačov úzko spojený, pretože sa počas svojho vývoja vzájomne ovplyvňovali. Jazyk je aplikáciou skôr vyvinutého rozsiahleho značkovacieho jazyka SGML.

Za vznik prvotnej verzie jazyka v rokoch 1990-1991 je zodpovedný Tim Berners-Lee[13], vynálezca HTTP, URL, WWW a riaditeľ konzorcia W3C[14], ktoré sa ďalej stará o vývoj a štandardizáciu jazyka. Táto prvotná verzia nepodporovala grafický režim. Jednalo sa takisto o verziu, ktorá nebola oficiálne formálne špecifikovaná.
Nasledoval rýchly rozvoj webu, preto bolo nutné pre HTML definovať štandardy. Nebudeme sa zaoberať už nepoužívanými verziami HTML 2.0 a 3.2 ale pozrieme sa rovno až na aktuálne najdôležitejšie verzie jazyka.

HTML 4.0

Verzia 4.0 bola vydaná v roku 1997. Špecifikácia jazyka bola oproti predchádzajúcej verzií rozšírená o nové prvky pre tvorbu tabuliek, formulárov a pribudli aj štandardizované rámy (frames). Táto verzia sa snaží dosiahnuť pôvodný cieľ, aby prvky vyznačovali význam (sémantiku) jednotlivých častí dokumentu. Vzhľad ma byť vytvorený pomocou pripojených štýlov. Niektoré prezentačné elementy boli z toho dôvodu zamietnuté.

HTML 4.1

Verzia 4.01 bola vydaná v roku 1999. Opravuje niektoré chyby predchádzajúcej verzie. Podľa predpokladov sa malo jednať o poslednú verziu pred prechodom na XHTML.

HTML verzie 4 definuje tri odlišné verzie jazyka. V originálnom znení sú to: *Strict*, *Transitional* (tiež nazývaná *Loose*) a *Frameset*. Verzia Strict je určená pre nové webové dokumenty a je považovaná za *„best practice“*. Verzie Transitional a Frameset boli vyvinuté kvôli jednoduchšej zmene dokumentov vyhovujúcich starším verziám jazyka HTML alebo nevyhovujúcich žiadnej špecifikácií jazyka. Tieto dve verzie umožňujú značkovanie prezenčnej vrstvy (ang. *„presentation markup“*), ktoré je vynechané vo verzii Strict. Namiesto toho je odporúčané používať CSS kaskádové štýly na tvorbu prezentačnej vrstvy

HTML dokumentov. CSS má v rámci našej práce dôležitý význam, čo zistíme neskôr v kapitole 4.3.

HTML 5.0

Verzia 5 začala vznikať v roku 2007, kedy bola založená nová pracovná skupina v rámci organizácie W3C, ktorej cieľom je vývoj novej verzie HTML. V súčasnosti (jún 2012) je podľa oficiálnych informácií na stránkach konzorcia W3C stále v štádiu návrhu (ang. „working draft") [15]. Zlepšením od verzie 4 majú byť skrátené a rýchlejšie zápisy. Dôraz je kladený na jednoduchosť a účinnosť zároveň.
V rámci tematiky tejto práce najzaujímavejším prínosom verzie 5 budú nové HTML značky sémanticky definujúce štruktúru stránky, čím budeme schopní od seba oddeliť jednotlivé časti webovej stránky. K tomu sa vrátime si povieme neskôr.

XHTML

V roku 2000 vznikol vďaka W3C rozšíriteľný značkovací jazyk XHTML. X na začiatku znamená rozšíriteľný (ang. „*e*Xtensible"). V skutočnosti sa jedná o zúženie a osekanie. Vychádza z jazyka HTML verzie 4, preto je mu veľmi podobný. Odlišuje sa najmä prísnejším syntaxom. Zatiaľ čo HTML je aplikáciou veľmi pružného značkovacieho jazyka SGML, XHTML je aplikáciu XML, obmedzenej podmnožiny SGML. Z toho vyplýva, že dokumenty XHTML sú súčasne dokumentmi XML a musia vyhovovať prísnym pravidlám jazyka XML.

Okrem spomínaného prísnejšieho syntaxu sa XHTML 1.0 od HTML 4.01 nelíši. Z toho vyplýva, že XHTML takisto obsahuje tri typy DTD - Strict, Transitional a Frameset. Až vo verzií XHTML 1.1 boli úplne odstránené niektoré elementy, ktoré sme mohli nájsť v Transitional a Frameset verziách (X)HTML. Jednotlivé tematicky spolu súvisiace prvky boli v jazyku XHTML zaradené do modulov.

4.2 Ako funguje HTML

O vývoji jazyka HTML sme sa vďaka predchádzajúcej kapitole dozvedeli dostatok informácií. Ostáva ešte odkryť praktické záležitosti ohľadom jazyka. Preto si veľmi stručne pripomenieme ako sa jazyk používa.

Základom dokumentu vytvoreného v značkovacom jazyku pre tvorbu WWW stránok je prostý text. Tak ako napr. Český jazyk obsahuje svoje slová, tak aj jazyk HTML má svoje slová. Nazývame ich „tagy" (značky). Označujeme nimi časti textu, čím im dávame špecifický význam a vlastnosti. HTML sémantické značky rozdeľujeme na párové a nepárové. Párové majú svoj začiatok a koniec.
Každý správny HTML dokument zachováva svoju základnú štruktúru. Jej podobu môžme vidieť vo výpise 2, ktorý poslúži ako vzorový príklad.
Prvý riadok hovorí, aká verzia HTML bola použitá. HTML dokument začína párovou značkou `<html> ... </html>`. Značky môžu ďalej obsahovať tzv. „atribúty", ktoré sú vždy súčasťou otváracej značky. Hodnota atribútu sa uzatvára do úvodzoviek (viz. príklad). V ukážkovom výpise môžme vidieť atribút `lang` priradený otváracej značke tela dokumentu. Jedná sa o informáciu o jazyku, v ktorom sa chystáme obsah stránok tvoriť (*„sk"* pre slovenčinu, *„cs"* pre češtinu).

V hlavičke dokumentu označenej `<head> ... </head>` sa umiestňujú informácie o stránke, ktorými sú meta značky a značka pre titulok stránky. Vo výpise vidíme meta značku označujúcu použité kódovanie.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.
    dtd">
<html lang="sk">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    < title > Titulok stránky </ title >
  </head>
  <body>
    ...
  </body>
</html>
```

Výpis 2: Základná štruktúra HTML dokumentu

Samotný obsah stránky sa uzatvára medzi značky `<body> ... </body>`. Obsah formujeme pomocou ďalších HTML značiek. K dispozícií máme rozsiahlu množinu značiek, ktoré môžme rozdeliť do nasledujúcich skupín: úprava textu, logické formátovanie, bloky, zoznamy, odkazy, obrázky, tabuľky, rámy, objekty, formuláre.

Značky pre konkrétne skupiny nebudeme uvádzať, tým pádom sa nebudeme ani zaoberať ich vysvetľovaním. V prípade potreby je možné navštíviť zdroj [16], z ktorého sa pri písaní tejto kapitoly vychádzalo a ktorý sa detailne venuje ako popisu HTML značiek tak aj jazyka samotného.

4.3 CSS Kaskádové štýly

Kaskádové štýly (CSS) je jazyk určený na vizuálne formátovanie internetových dokumentov napísaných v jazykoch HTML, XHTML alebo XML. Štýly umožňujú jednoduchým spôsobom oddeliť štruktúru dokumentu od prezentačnej vrstvy. Docielime tým prehľadný a jednoduchý kód. Jazyk bol navrhnutý a štandardizovaný organizáciou W3C.

Prvá verzia CSS 1 vznikla v roku 1996. Umožňovala len prácu s písmami, okrajmi a farbami. V roku 1998 vznikla verzia CSS 2, ktorá bola oproti CSS 1 doplnená o nové možnosti. V súčasnosti je podporovaná vo všetkých novších verziách webových prehliadačov. V roku 2011 bola dokončená revízia CSS 2.1. Konzorcium W3C momentálne pracuje na špecifikácií CSS 3.

Kaskádové štýly definujeme podľa špecifických pravidiel. Každé z týchto pravidiel obsahuje *selektor* a blok deklarácií. Blok deklarácií ďalej obsahuje zoznam deklarácií oddelených bodkočiarkou. Každá deklarácia pozostáva s identifikátoru vlastností nasledovaným dvojbodkou a hodnoty vlastností. Uveďme si jednoduchý schematický príklad pravidla:

```
selektor {
    identifikátor_vlastnosti:   hodnota_vlastnosti;
}
```

Výpis 3: CSS Pravidlo

CSS definuje niekoľko rôznych selektorov, ktoré budú mať v našej práci dôležitý význam, čo zistíme v nasledujúcich kapitolách. Treba však upozorniť, že pojem selektor budeme neskôr používať v inom význame ako je tomu pri CSS! Pre nás najväčší význam majú konkrétne dva z nich.
Prvý z nich sa zapisuje formou `.trieda` a platí pre všetky elementy HTML kódu, ktoré majú atribút `class` nastavený na trieda, teda `<tag class="trieda">`.
Druhým dôležitým selektorom je `#identifikátor`, ktorý platí pre všetky elementy, ktoré majú atribút `id` nastavený na `identifikátor` , teda:
`<tag id="identifikátor">`.

Tvorba prezentačnej vrstvy webových stránok pomocou HTML a CSS, konkrétne rozloženie jednotlivých častí stránky akými sú hlavička, menu atď., sa v dnešnej dobe realizuje vo väčšine prípadov pomocou HTML značiek pre oddiel. Reč je o párovej HTML značke `<div>`. Zahrňuje v sebe ľubovoľne veľkú oblasť textu, vrátane nadpisov, obrázkov a tabuliek. Táto značka bola do HTML pridaný najmä kvôli kaskádovým štýlom, ktoré umožňujú nastaviť formátovacie vlastnosti oddielu podľa našich predstav. Podobne je to aj so značkou `<p>`, ktorá je používaná najmä pre textové odseky.
To má úzku spojitosť s CSS selektormi, pretože formátovacie vlastnosti oddielom a textovým odsekom priraďujeme práve pomocou nich. Pre hlavičku webovej stránky tak môže vzniknúť napr. oddiel `<div class="header">` alebo analogicky s využitím druhého selektoru `<div id="header">`.

4.4 HTML formát spravodajských serverov

Aké verzie HTML jazyka sú používané v rámci českých spravodajských serverov? Pri teoretickom výklade sa budeme zaoberať množinu šiestich aktuálne najpolárnejších spravodajských serverov v rámci Českej republiky. Najpopulárnejšími spravodajskými servermi sú samozrejme myslené najnavštevovanejšie.

Popularita bola určená vďaka projektu „NetMonitor.cz“ [17] na základe verejnej správy za obdobie apríl 2012. NetMonitor je rozsiahly výskumný projekt, ktorého cieľom je poskytnúť informácie o návštevnosti internetu a sociodemografickom profile jeho návštevníkov v Českej republike.
Podľa tejto aprílovej správy sa do množiny šiestich najnavštevovanejších spravodajských serverov v rámci kategórie spravodajstvo dostali (zostupne od najnavštevovanejšieho, zapísané v rovnakej forme ako je uvedené v správe):

1. novinky.cz | rubrika Zpravodajství (HTML 4)
2. idnes.cz | Zprávy (HTML 4)
3. centrum.cz | Aktualne.cz | Zprávy Aktualne.cz (XHTML 1.0 Transitional)
4. denik.cz | Zpravodajství Deník (XHTML 1.0 Strict)
5. nova.cz | TN.CZ | tn_ZPRAVODAJSTVÍ (XHTML 1.0 Transitional)
6. lidovky.cz | Zpravodajství (HTML 4)

Súčasťou zoznamu je priamo uvedená aj verzia použitej verzie jazyka HTML resp. XHTML. *Na základe skúmania aj ostatných serverov môžeme uviesť, že HTML verzie 4 a XHTML verzie 1.0 sú najčastejšími verziami HTML formátu používaného českými spravodajskými servermi.*
Na začiatku kapitoly sme sa dozvedeli, že rozdiely medzi týmito verziami sú naozaj veľmi malé, preto sa nemusíme ďalej zaoberať použitou verziou jazyka HTML.

4.5 Text článku a jeho identifikácia

V otázke HTML formátu českých spravodajských serverov už máme jasno, pozrime sa preto na problémy spojené so strojovým spracovaním spravodajských článkov. Aby sme mohli vykonávať takú špecifickú operáciu akou je monitorovanie frekvencie slov na jednotlivých spravodajských serveroch, musíme byť najskôr schopný určiť , kde na stránke sa hlavný text článku vlastne nachádza.

Ešte predtým si však uvedomme, z čoho sa skladá samotný článok. *Článok (resp. text, ktorý nás zaujíma) sa skladá z jeho nadpisu, prvého informačného odseku a hlavného tela článku.* Nie je to však pravidlom.
Prvý informačný odsek textu, väčšinou označovaný aj ako *„perex“*, je výnimočne na niektorých serveroch súčasťou hlavného textu článku. Príkladom takéhoto spravodajského

serveru môže byť „aktuálne.cz“ prítomne v našej modelovej množine serverov. Na základe podrobných sledovaní sa ukázalo, že tento server je jedným z množiny špecifických serverov, ktoré neodlišujú prvý informačný odsek svojich článkov od hlavného textu článku. To sa nám samozrejme prejaví v neskoršej časti práce.

Každý článok je súčasťou webovej stránky uverejnenej na danom serveri. Súčasťou každej spravodajskej stránky je však okrem vlastného textu článku aj mnoho ďalších prvkov, ktoré nám robia strojové extrahovanie textu náročným, pretože musíme byť schopní určiť, čo patrí k článku a čo nie. Medzi takéto prvky patria najmä reklamy ale typicky aj navigácia, odkazy na podobné články a pod.

Ideálnym riešením tohto problému by bol prechod na jazyk HTML verzie 5. Ako sme spomínali v kapitole 4.1, jedným s najzaujímavejších prínosov verzie 5 sú nové HTML značky sémanticky definujúce štruktúru stránky. Pomocou týchto značiek budeme schopní v zdrojovom kóde stránky presne označiť čo je hlavička a pätička stránky, ktorá časť kódu tvorí menu stránky a takisto budeme mať možnosť presne označiť hlavný text stránky (v oblasti internetovej žurnalistiky hlavný text článku). Tým by bol náš problém vyriešený, pretože by sme mali k dispozícií univerzálny identifikátor textu článku, ktorý je ohniskom nášho záujmu.

Vráťme sa späť do súčasnosti, kedy jazyk HTML verzie 5 je stále v štádiu vývoja. Ako sme sa už dozvedeli, väčšina serverov má svoju prezenčnú vrstvu postavenú na HTML verzie 4, poprípade XHTML. Tie neobsahujú špecifické značky sémanticky definujúce štruktúru stránky, ktoré by nám umožnili jednoducho vyriešiť náš problém s identifikáciou hlavného textu.

Základným stavebným kameňom všetkých týchto webových stránok je odsek, označený pomocou párovej značky `<div>`. Každý odsek má pomocou kaskádových štýlov nastavené špecifické formátovacie vlastnosti. Aby konkrétny `<div>` vedel, aké vlastnosti mu prislúchajú, využíva atribúty `id` alebo `class`, ktoré jednoznačne definujú príslušnú množinu CSS vlastností (detaily v kapitole 4.3). Môžme sa na to pozerať ako na momentálnu alternatívu sémantických značiek v rámci HTML 5. Pomocou týchto odsekov sme takisto schopní určiť jednotlivé časti stránky: hlavičku, pätu, menu ale najmä samotný článok.

Oproti očakávanej najnovšej verzie jazyka HTML má však tento systém „sémantických“ odsekov jednu závažnú nevýhodu. Neexistuje totiž formálne špecifikovaná množina pre názvy atribútov `class` alebo `id`. Voľba názvov je ponechaná na voľbe autora stránok. Avšak je zachovaná jednotnosť zvolených názvov atribútov v rámci jedného serveru. To nám dáva možnosť pomocou týchto odsekov identifikovať text článku na danom serveri, avšak očakávajme, že na inom serveri budú identifikované iné odseky.

4.6 Predpoklady podporovaného serveru

Z doterajšieho výkladu vyplýva množina dôležitých predpokladov, ktorú si rozdelíme na tzv. „predpoklady podporovaného serveru“ a „predpoklady korektnej extrakcie“. Je to množina požiadaviek, ktoré musia byť splnené, aby mohol byť server pridaný do spracovania našim nástrojom. V niektorých prípadoch sa jedná o predpoklady čisto teoretické, ktorých porušenie nebolo prakticky nikdy overené, pretože ich spĺňajú všetky servery, s ktorými sa pracovalo.

1. *Český internetový spravodajský server, ktorý chceme sledovať, musí poskytovať RSS informačný/é kanál/y, slúžiaci/e ako zdroj článkov.* Môže sa jednať o všeobecný kanál pre všetky prichádzajúce správy alebo o kanál čiastkový. V prípade existencie čiastkových kanálov, vzťahujúcim sa k jednotlivým kategóriám, je možné sledovať konkrétne kategórie správ.
   Pri absencii RSS informačných kanálov nie je možné sledovať daný server.

2. Články v RSS kanále musia obsahovať nevyhnutné elementy. Sú nimi `<link>`, `<title>` a `<description>`. Element `<pubDate>` je nahraditeľný. Pri jeho absencii je možné generovať systémový dátum.

4.7 Predpoklady korektnej extrakcie

Ak sú splnené nutné predpoklady pre pridanie serveru do monitorovania, je potrebné aby bola splnená aj ďalšia množina predpokladov, ak chceme, aby boli nastavené správne selektory, podľa ktorých bude následne vyextrahovaný text článku.

1. Požadovaná je „div“ štruktúra zdrojového kódu, pričom tieto značky sú špecifikované buď atribútom `class` alebo `id` s príslušnými hodnotami.

2. Validný HTML kód do takej úrovne, aby neovplyvnil činnosť programu pri hľadaní textu článku.

3. Prvý informačný odsek textu článku je uvedený v samostatnom `<DIV>` elemente, čím môžeme jednoznačne určiť miesto jeho výskytu v zdrojovom kóde. V prípade, že je prvý informačný odsek súčasťou samotného textu článku, sa text článku vyextrahuje správne, pričom bude použitý iba jeden selektor.

4. Významový text uložený v zodpovedajúcom elemente `<description>` v zdrojovom kóde RSS kanálu, sa musí zhodovať z textom prvého informačného odseku textu článku.

5. Samotný text článku je celý uzavretý v `<DIV>` elemente, čím môžeme jednoznačne určiť miesto jeho výskytu v zdrojovom kóde. V prípade, že samotný text článku je po častiach rozdelený a uzavretý v rôznych `<DIV>` elementoch, môže byť v závislosti od ďalších okolností, vyextrahovaná len časť textu alebo žiaden text.

6. V článku sa nachádza odsek textu, ktorý obsahuje aspoň dve vety s počtom tri slová, aby bolo možné podľa zvolenej masky identifikovať element <DIV>, ktorý obsahuje text článku. Pri porušení podmienky a vzhľadom na ďalšie okolnosti, môže byť vyextrahovaný nevyhovujúci (iný text v zdrojovom kóde vyhovujúci maske) alebo žiaden text.

7. Pri splnení predchádzajúcej podmienky musí platiť, že dĺžka obsahového textu identifikovaného elementu je najväčšia v rámci všetkých ostaných elementov <DIV>, ktoré vyhovujú zadanej maske. Toto platí pri väčšine prichádzajúcich správ aktuálne zapísaných v RSS kanále, aby bol výsledný selektor správny.

Uvedené predpoklady konkrétnej extrakcie textu veľmi úzko súvisia s algoritmom pre vyhľadávanie selektorov uvedenom v kapitole 6.2

5 Špecifikácia požiadaviek

V tejto kapitole sa budeme venovať špecifikácií funkčných požiadaviek systému, aby sme získali ucelený obraz o tom, aké funkcie má systém spĺňať.
Pre špecifikáciu funkčných požiadaviek použijeme jednu z najjednoduchších foriem zápisu. Požiadavky budeme organizovať do skupín základných funkcií (ang. „features“). K ním sa ďalej rozpisujú podrobné požiadavky (ang. „requirements“).

Požiadavka je schopnosť, ktorú produkt musí poskytovať alebo niečo, čo produkt musí robiť, aby v konečnom dôsledku uspokojil zákaznícku potrebu.
Vlastnosť/funkcia je množina súvisiacich požiadaviek, ktoré umožňujú užívateľovi uspokojiť biznis cieľ alebo potrebu.

Funkcia 1 - Manažment RSS kanálov

Požiadavka 1.1 Užívateľ (administrátor) má možnosť pridať nový RSS kanál do spracovania, pričom môže využiť manuálnu, alebo automatickú metódu pridávania selektorov. Pri automatickej metóde sa systém pokúsi vyhľadať a nastaviť selektory pre RSS kanál automaticky na základe navrhnutého algoritmu. Manuálna metóda je závislá od schopnosti užívateľa porozumieť teórií selektorov, aby bol schopný nastaviť ich manuálne.
Požiadavka 1.2 Užívateľ (administrátor, verejnosť) má možnosť prezerať si zoznam pridaných RSS kanálov.
Požiadavka 1.3 Užívateľ (administrátor) môže v prípade potreby upraviť selektory jednotlivých RSS kanálov alebo zmazať celý RSS kanál. So zmazaním RSS kanálu budú zmazané všetky slová a články prislúchajúce k nemu.
Požiadavka 1.4 Ak je monitorovanie RSS kanálov spustené, systém je po pridaní nového RSS kanálu schopný okamžite spustiť spracovanie článkov pre tento novo pridaný RSS kanál a následne ho priradiť do monitorovania.

Funkcia 2 - Manažment stop slov

Požiadavka 2.1 Užívateľ (administrátor) má možnosť pridávať nové stop slová do jednoduchého zoznamu stop slov. Takisto ich môže svojvoľne zo zoznamu mazať.
Požiadavka 2.2 Systém po vložení nového stop slova aktualizuje zoznam doposiaľ zaznamenaných slov a ich frekvencií. Všetky slová, ktoré boli označené ako stop slová budú zmazané.
Požiadavka 2.3 Systém pri spracovaní článkov porovnáva jednotlivé slová so zoznamom stop slov a automaticky odstraňuje stop slova zo spracovania.

Funkcia 3 - Manažment ekvivalentných slov

Požiadavka 3.1 Užívateľ (administrátor) má možnosť vytvárať zoznam nových ekvivalentných skupín slov.
Požiadavka 3.2 Užívateľ (administrátor) môže upravovať jednotlivé tvary slov, alebo mazať celé ekvivalentné skupiny.
Požiadavka 3.3 Systém po vložení novej ekvivalentnej skupiny aktualizuje zoznam do-

posiaľ zaznamenaných slov a ich frekvencií. Všetky slová, ktorým bol priradený koreň, budú prevedené na tento koreň.
Požiadavka 3.4 Systém pri spracovaní článkov porovnáva jednotlivé slová so zoznamom ekvivalentných slov a automaticky prevádza slova na ich koreň podľa slovníku ekvivalentných slov.

Funkcia 4 - Zber štatistických údajov

Požiadavka 4.1 Systém je schopný pracovať s jednotlivými článkami uvedenými v zadaných RSS kanáloch (RSS čítačka). Dokáže z nich efektívne zhromažďovať nielen štatistiky o frekvencií výskytu jednotlivých slov za určité časové obdobie, ale aj jednotlivé články a dokáže určiť dôležitosť jednotlivých udalosti.
Požiadavka 4.2 Pri spracovaní jednotlivých slov systém automaticky vyraďuje stop slová a vykonáva proces lematizácie s využitím slovníku ekvivalentných slov.
Požiadavka 4.3 Systém dokáže rozpoznať už spracované články a zabrániť ich redundantnému spracovaniu.
Požiadavka 4.4 Po spustení monitorovania systém dokáže v pravidelných časových intervaloch opätovne kontrolovať RSS kanály, hľadať a spracovávať nové články.
Požiadavka 4.5 Systém automaticky ukladá spracované slová do perzistentného dátového úložiska a počíta ich frekvenciu pre jednotlivé dni.

Funkcia 5 - Zobrazovanie štatistických údajov

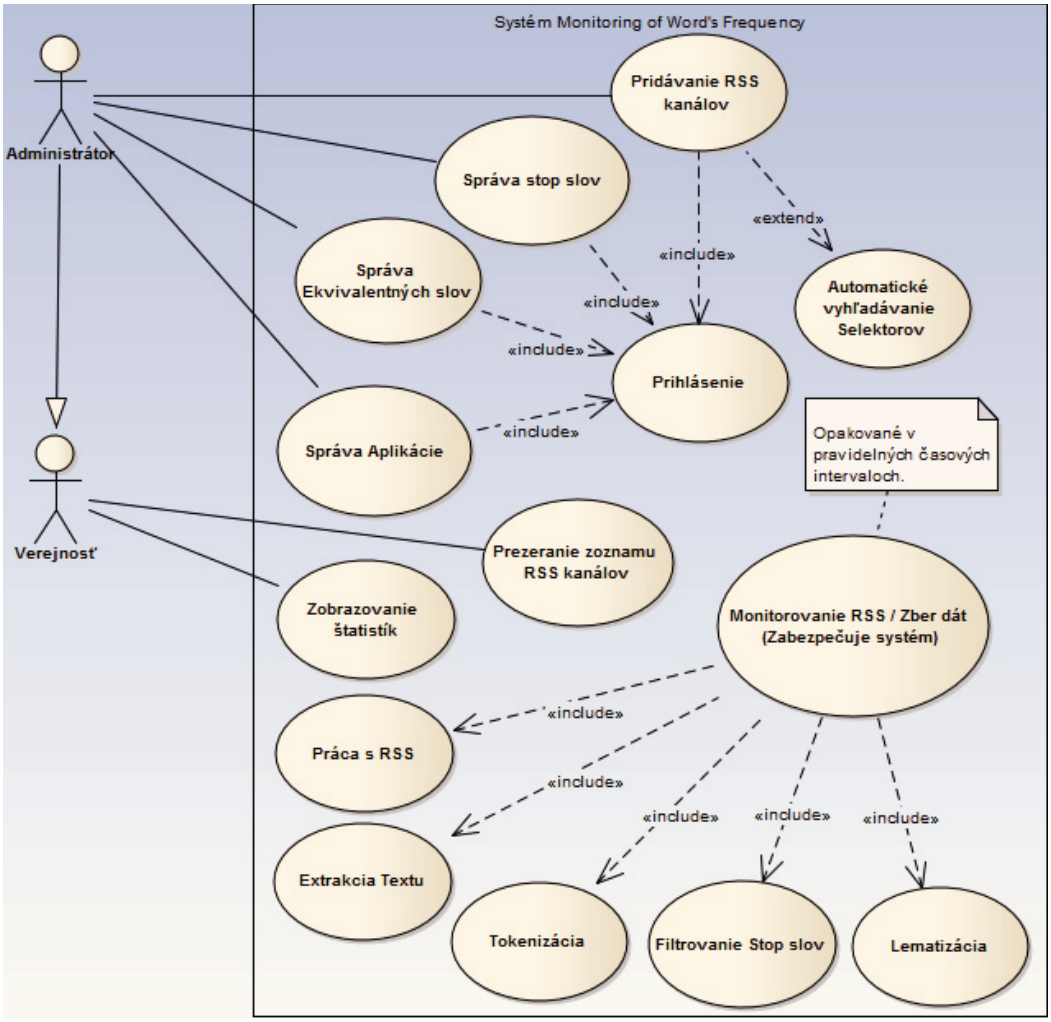
Požiadavka 5.1 Užívateľ (administrátor, verejnosť) má možnosť zobrazovať si zozbierané štatistiky slov a jednotlivých článkov.
Výsledné štatistické zoznamy môžu byť špecifikované podľa času, serveru a kategórie.
Požiadavka 5.2 Užívateľ (administrátor, verejnosť) môže využiť možnosť vyhľadávania v článkoch podľa kľúčových slov.

Funkcia 6 - Manažment aplikácie

Požiadavka 6.1 Užívateľ (administrátor) má možnosť spustiť alebo zastaviť monitorovanie RSS kanálov (zber dát).
Požiadavka 6.2 Užívateľ (administrátor) má možnosť nastaviť časový interval opätovného spracovávania RSS kanálov z dôvodu kontroly nových článkov.
Požiadavka 6.3 Systém je schopný zaznamenávať najdôležitejšie informácie ohľadom behu aplikácie do informačných logov.
Požiadavka 6.4 Užívateľovi (administrátor) je umožnené sledovať informačné logy v rámci manažmentu aplikácie.

Najdôležitejšie užívateľské ciele zachytíme pomocou Use Case diagramu na obrázku 3. V systéme vystupujú dve užívateľské role:

- Administrátor
- Verejnosť



Obrázok 3: Use Case Diagram Systému

Z diagramu je zrejmé, že administrátor má absolútnu právomoc v rámci celého systému, oproti verejnému užívateľovi, ktorý si môže zobrazovať len zoznam spracovávaných RSS kanálov a štatistík.
Za zmienku stojí prípad užitia Monitorovanie RSS kanálov, ktorý zabezpečuje zber dát. Celý tento proces je riadený systémom a bude spúšťaný v pravidelných časových intervaloch, aby bolo možné sledovať a spracovávať nové články.

6 Analýza a návrh

Cieľom tejto kapitoly je podať ucelený pohľad na systém ako celok a takisto jeho jednotlivých častí. Máme na mysli pohľad „zhora“ na cele naše riešenie.
V predchádzajúcej kapitole sme získali prvotné informácie o funkciách, ktoré sú od systému očakávané. Z týchto informácií sme schopný vytvoriť prvotný „hrubý“ náhľad na systém ako celok (obr. 4).

Z diagramu je zrejmé, že systém sa bude skladať z troch základných programových modulov.

1. Samotný informačný systém, ktorého úlohou bude poskytovať rozhranie pre komunikáciu s užívateľom.
2. Druhou veľkou časťou bude RSS čítačka, ktorá bude mať na starosti prácu so zdrojom článkov (RSS kanály).
3. Poslednou časťou systému bude extraktor textu článkov z HTML stránok, ktorý bude takisto poskytovať metódu pre automatické vyhľadávanie selektorov pre jednotlivé RSS kanály.

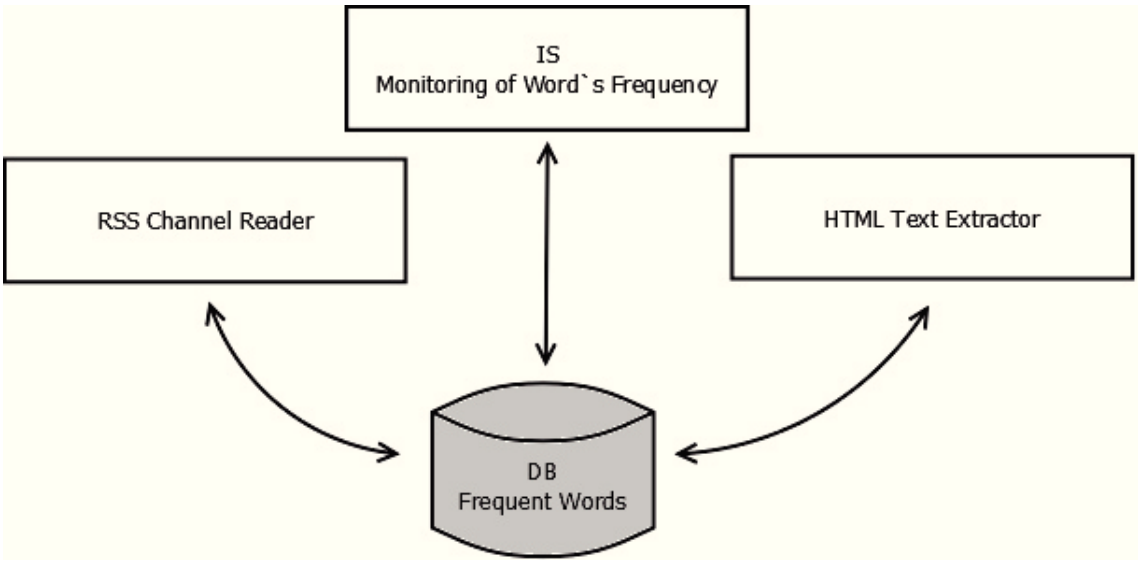
Okrem toho nesmieme zabudnúť na veľmi dôležitú časť systému, ktorou bude databáza slúžiaca na uchovávanie dát. V tejto kapitole sa preto budeme takisto venovať návrhu dátového úložiska. Vidíme, že jednotlivé programové moduly sú na sebe viac menej nezávislé a dáta si predávajú prostredníctvom databázy.

Statickú štruktúru systému v analytickej verzii si znázornime pomocou triedneho diagramu UML (obr. 5), ktorý podáva ucelený náhľad na budúce triedy celého systému.

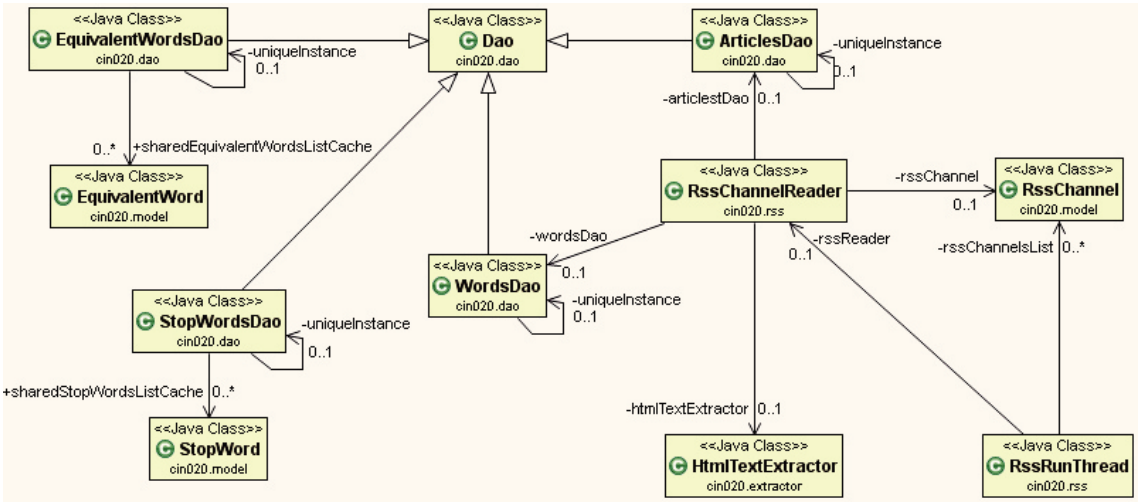
Ďalej si popíšeme jednotlivé časti, z ktorých je systém zložený. Detailne si popíšeme, ako sa postupovalo pri analýze a návrhu riešení jednotlivých problémov a ktoré techniky boli použité.

6.1 Analýza RSS kanálov

Prvým krokom pri návrhu systému bolo vyriešiť analýzu RSS kanálov. Pretože RSS je podmnožina XML, hovoríme o *„parsovaní RSS kanálov“*. V kapitole 3 sme sa zaoberali technológiou RSS. V rámci nej sme sa zamerali najmä na dôležité elementy popisujúce články, ktoré potrebujeme z RSS kanálu získať a následne s nimi pracovať.
Jednoducho povedané, chceme navrhnúť časť programu, ktorá umožni načítavať články z RSS kanálu spolu s potrebnými informáciami o nich. Touto časťou bude „špeciálna RSS čítačka“.



Obrázok 4: Architektúra systému



Obrázok 5: Analytický UML Class Diagram

6.1.1 RSS čítačka

Chceme navrhnúť funkčnosť špeciálnej RSS čítačky s cieľom získavania potrebných informácií o článkoch z informačných RSS kanálov.
Čítačka po spustení sekvenčne prejde zoznam všetkých evidovaných RSS kanálov a všetkých článkov v nich. V rámci článkov sa zisťuje názov, popis, URL, dátum publikácie a jednoznačný textový identifikátor článku. Po prvotnom spracovaní kanálu sa dokáže na kanál znovu pripojiť v zvolenom časovom intervale (implicitne 30 minút) a zisťovať prítomnosť nových článkov od posledného spracovania.

Pri detekcii nových článkov využívame dva dôležité kroky zabraňujúce redundantnému spracovaniu tých istých článkov. Prvým z nich je zavedenie tzv. „posledného prečítaného článku“. V kapitole 3 sme si predstavili najdôležitejšie elementy popisujúce každý článok v RSS kanále. Jedným z nich je element `<guid>`. Uzatvára reťazec jednoznačne identifikujúci článok. To je presne to, čo potrebujeme, aby sme vedeli od seba odlíšiť jednotlivé články.
Princíp je jednoduchý. Spolu s RSS kanálom si vždy pamätáme identifikačný reťazec posledného prečítaného článku z tohto kanálu (ako súčasť záznamu v databáze). Po opätovnom spracovaní RSS kanálu z dôvodu kontroly nových článkov, využijeme identifikátor posledného prečítaného článku. Pretože články sú v RSS usporiadané vo forme frontu (kapitola 3.2), v cykle prechádzame všetky články, až narazíme na ten, ktorý bol naposledy spracovaný. Všetky nasledujúce články považujeme za nové a začneme ich spracovávať.

Ako sme si povedali v teoretickej kapitole o RSS, môže nastať situácia, že do RSS kanálu je znovu vložený článok, ktorý v ňom už bol, ale z dôvodu dôležitosti a propagácie, bol zaradený znovu. Tento problém je vyriešený takisto pomocou hodnoty elementu `<guid>`. Perzistentne si pamätáme všetky tieto hodnoty, s ktorými následne porovnávame identifikačné hodnoty nových článkov. Ak sa rovnajú, znamená to, že článok bol už spracovaný a prejdeme na ďalší v poradí bez jeho opätovného spracovania.

V skutočnosti by sme si vystačili len s využitím porovnávania identifikačných hodnôt článkov. Zavedenie posledného prečítaného článku ale umožňuje zrýchlenie spracovania RSS kanálov.

Dynamické chovanie tried je zachytené pomocou sekvenčných diagramov na obrázkoch 6 a 7.
Prvý z nich znázorňuje základné vlákno čítačky, ktoré pokiaľ má právo k činnosti (aplikácia je v stave „bežiaca“), zostrojí zoznam RSS kanálov, nastaví RSS čítačku, ktorú bude používať a sekvenčne začne predávať jednotlivé RSS kanály k spracovaniu. Po spracovaní posledného kanálu sa vlákno uspí na zvolený časový interval (implicitne 30 minút) a pokiaľ nestratí právo k svojej činnosti (monitorovanie sa dostane vonkajším podnetom do stavu „zastavené“) opakuje automaticky tento proces v pravidelných časových intervaloch.

Zvláštnosťou tohto vlákna je, že ak dostane na vstupe zadaný konkrétny RSS kanál, spracuje len tento jeden kanál a svoju činnosť následne ukončí. To využijeme pri pridávaní nových RSS kanálov, kedy je po jeho pridaní požadované okamžité spracovanie kanálu. V opačnom prípade sa začne spracovávať kompletný zoznam pridaných informačných kanálov, ako to bolo popísane vyššie.

Druhý sekvenčný diagram popisuje proces spracovania konkrétneho RSS kanálu. Úvodom sa nastavuje identifikátor posledného spracovaného článku prislúchajúcemu k tomuto kanálu. Následne sa vyhľadáva pozícia naposledy spracovaného článku v RSS kanále. V prípade, že tento článok sa v kanále už nenachádza (bol vytlačený novými článkami), alebo sa jedna o prvý krát spracovávaný kanál, začne sa sekvečné spracovanie všetkých článkov. V opačnom prípade sa pokračuje v spracovávaní článkov od posledného spracovaného.
Ďalej bude nasledovať práca s jednotlivými článkami v RSS kanále.

6.2 Vyhľadávanie selektorov

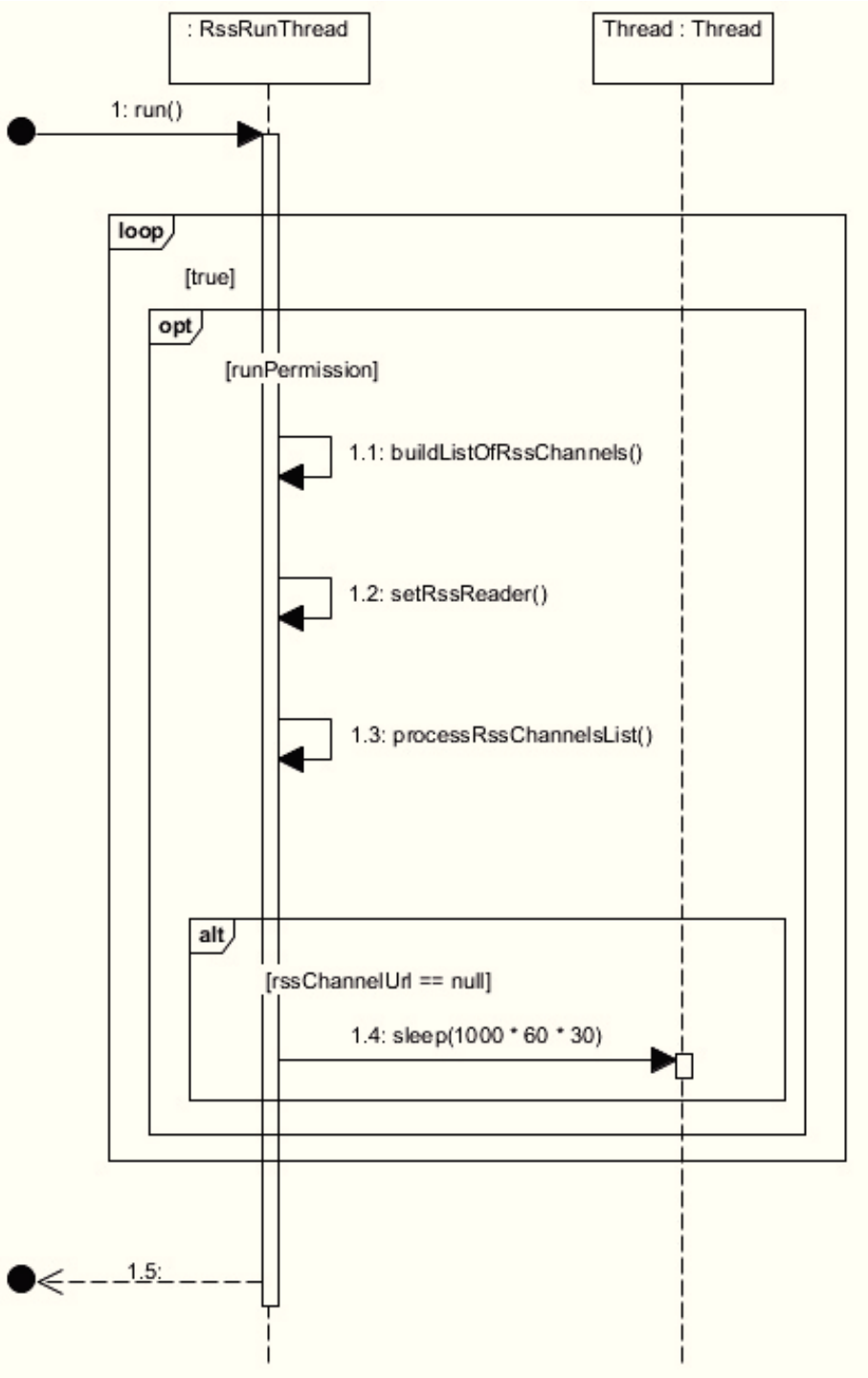
V kapitole 4.5 sme vyriešili identifikáciu textu článku s využitím HMTL značiek a CSS selektorov. Aby sme mohli text týmto spôsobom identifikovať, zostáva stále nezodpovedaná jedna veľmi dôležitá otázka: *„Čo je vlastne samotným textom článku?“* Teda ak chcem identifikovať umiestenie textu článku v štruktúre HTML kódu, musíme najskôr identifikovať samotný text, ktorý ho tvorí. Keď sa pozrieme na ľubovoľný článok na niektorom zo spravodajských serverov, je nám ihneď jasné, čo patrí do jeho obsahu a čo nie. Počítač bohužiaľ takéto „jednoznačné videnie“ nemá a preto treba požiť iný prístup. K tomuto účelu sa dá využiť niekoľko strojových metód, ktoré využijeme pre tento účel.

Jednou zo základných a zároveň najdôležitejších častí práce je programový modul, ktorý dokáže určiť a skonštruovať špecifické selektory. **Tu sa dostávame k pravému významu pojmu selektor používanom v tejto práci. Pod týmto termínom si môžeme predstaviť syntakticky presne daný zápis, ktorý jednoznačne určuje umiestenie obsahového textu článku v rámci jedného RSS kanálu.** Tie budú slúžiť ďalej ako vstup pre nástroj na extrakciu textu, schopný vyextrahovať podľa nich text článku. V kapitole 7.2, je uvedená aj ich presná podoba a pravidla pre vytváranie selektorov.

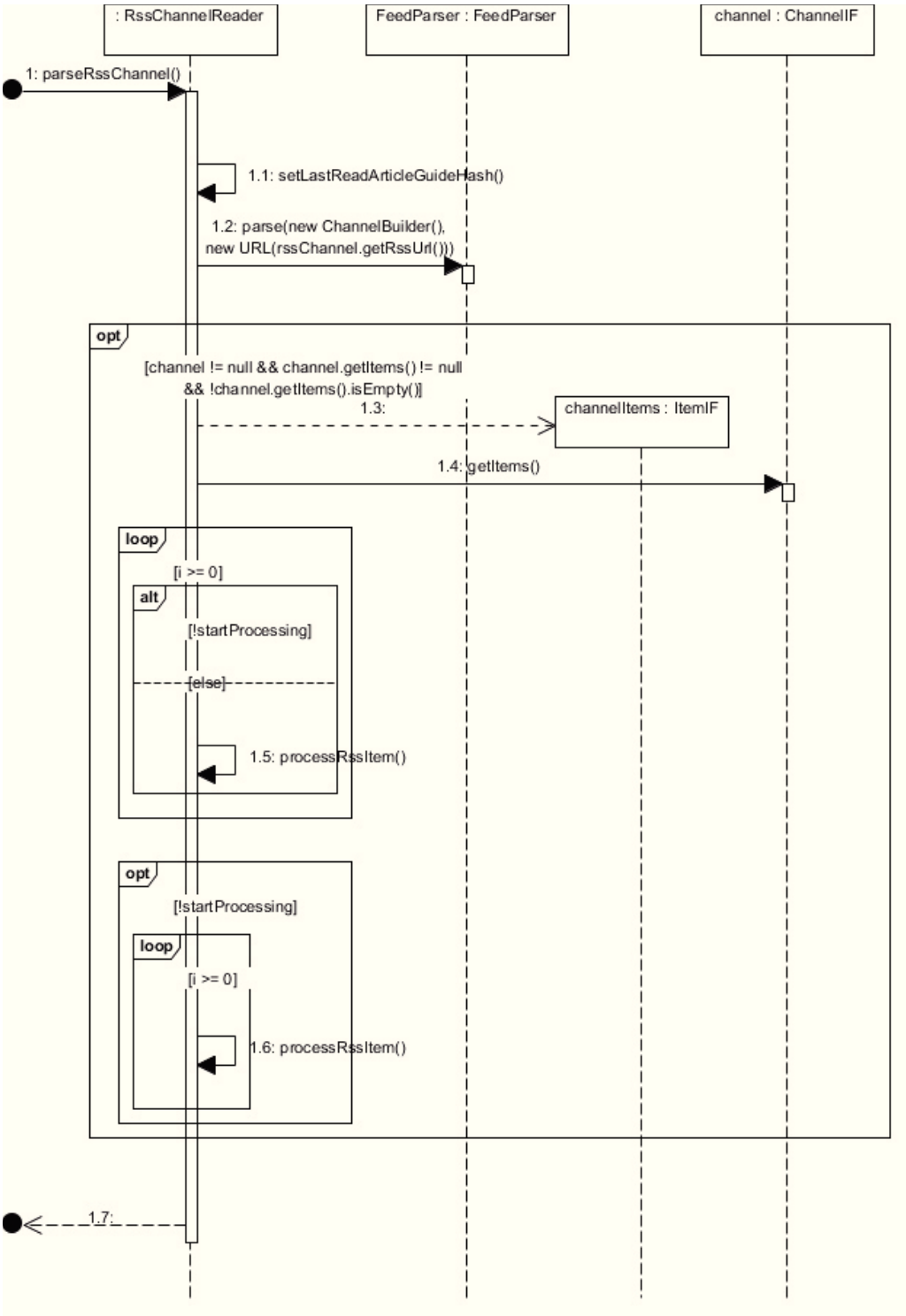
HTML štruktúra spravodajských článkov sa na jednotlivých serveroch značne líši. Na základe všetkých spoločných aj odlišných znakov bol vytvorený komplexný algoritmus, ktorý sa snaží byť čo najviac univerzálnym a korektne pracovať s čo najväčším počtom informačných kanálov na rôznych spravodajských serveroch. Popíšme si podrobne ako tento algoritmus funguje.

6.2.1 Stratégia postupu

Hľadáme množinu selektorov, ktoré následne umožnia extrahovať text článkov. Vieme, že nie sú rovnaké pre všetky spravodajské servery resp. im náležiace RSS kanály. Algo-



Obrázok 6: RssRunThread UML Sekvenčný diagram



Obrázok 7: RssChannelReader UML Sekvenčný diagram

ritmus, ktorý dostane na vstupe URL adresu RSS kanálu patriaceho nejakému českému spravodajskému serveru, musí byť schopný určiť selektory pre články obsiahnuté v tomto kanále. Na začiatku je dôležité uvedomiť si skutočnosť, že *určovať selektory pre hocijaký RSS kanál na základe jedného článku je síce možné, ale nie správne riešenie.*

Predstavme si, že sa rozhodneme určovať selektory na základe prvého resp. ľubovoľného článku z informačného kanálu. Výhodou takéhoto prístupu by bola rýchlosť, pretože vyhľadanie selektorov v jednom článku podľa vytvoreného algoritmu je relatívne časovo nenáročná operácia. *Môže však dôjsť k situácií, že selektory na základe tohto článku nebudú nastavene správne.* Typicky sa môže jednať o článok z kategórie šport, ktorý nemusí obsahovať žiaden významový text. Obsahom takýchto článkov môžu byť rôzne tabuľky športových výsledkov, diskusie alebo video reportáže. K tomu môže dôjsť aj vďaka rôznym ďalším „netradičných" článkom.

Preto bolo zvolené oveľa lepšie riešenie. *Identifikujeme a skonštruujeme selektory zvlášť pre každý článok, ktorý sa v danom čase nachádza v RSS kanále.* Z výsledného zoznamu nájdených selektorov určíme množinu, ktorá sa tam nachádza najčastejšie, spolu s percentuálnym číselným vyjadrením jej výskytu. Môžeme tak tvrdiť, že v danom čase pre zvolený RSS kanál sme našli množinu selektorov s danou percentuálnou pravdepodobnosťou korektnosti. Od predchádzajúceho riešenia sa jedná o časovo oveľa náročnejší proces závisiaci najmä na počte článkov v RSS kanále. Riešenie nám však umožňuje identifikovať selektory správne aj v prípade, že obsahom kanálu sú aj „netradičné" články, čo nám zabezpečuje vyššiu mieru presnosti.

6.2.2 Algoritmus

V tejto kapitole si čo najzrozumiteľnejšie popíšeme navrhnutý algoritmus pre identifikáciu selektorov v rámci jedného článku (jednej webovej stránky). Nasledovať budú jednotlivé kroky algoritmu spolu s ich popisom a vysvetlením. Zvolený bol textový popis algoritmu, pretože je veľmi dôležité vysvetliť jednotlivé kroky a dôvody, ktoré nás viedli k ich použitiu. Budeme sa držať na istej úrovni granularity. Nepôjdeme až do najmenších detailov algoritmu, pretože tie nie sú pre jeho vysvetlenie a pochopenie dôležite.

Na vstupe požaduje základné údaje o článku získané z RSS kanálu. Konkrétne sa jedná o URL adresu článku, aby sme mohli získať jeho zdrojový kód. Ďalej potrebujeme časť prvého informačného odseku článku získaného z RSS časti `description` , aby sme mohli identifikovať v zdrojovom kóde začiatok textu článku. Na to nám stačí niekoľko prvých znakov tohto odseku. Podľa nich jednoducho nájdeme zhodu v zdrojovom kóde, čím určíme pozíciu prvého odseku článku. V nasledujúcich 8 krokoch si popíšeme priebeh algoritmu:

1. **Z hlavičky stránky zisti jej kódovanie.**
2. **Na základe zisteného kódovania získaj zdrojový kód.**

3. **Zo zdrojového kódu odstráň hlavičku.**
   Dôvodom pre odstránenie hlavičky je fakt, že sa v nej už nenachádzajú dôležité údaje, preto aj z hľadiska optimalizácie je užitočné hlavičku zo zdrojového kódu odstrániť. Hlavička článku takisto môže obsahovať metadáta, ktorých obsahom býva ako nadpis, tak aj prvý informačný odsek článku. Jej odstránením sa zbavíme duplicít, ktoré by nám znemožnili správnu funkčnosť algoritmu.

4. **Vyčistí zdrojový kód od nepotrebných HTML elementov (aj s ich obsahom).**
   Zdrojový kód článku obsahuje množstvo HTML tagov, ktoré sú v rámci identifikácie selektorov nepotrebné, dokonca môžu spôsobiť nesprávne nastavenie selektorov. Tým, že sa ich zbavíme spolu s ich obsahom, očistíme zdrojový kód od nežiaducich a zavádzajúcich častí. Konkrétne sa jedná o tieto HTML značky:

   - Komentáre
   - Odkazy
   - Java skripty
   - Tabuľky
   - Obrázky
   - Logicky vymedzená časť textu (SPAN)

5. **Na základe časti textu prvého informačného odseku článku z RSS, nájdi jeho pozíciu v zdrojovom kóde. Ak sa v zdrojovom kóde nachádza, zistí ktorý oddiel (`div`) alebo odsek (`p`) bezprostredne pokrýva celý tento text a skonštruuj z neho selektor podľa príslušných pravidiel.**

6. **Podľa regulárneho výrazu označujúceho dve vety s aspoň tromi slovami, vyhľadaj všetky nálezy v zdrojovom kóde.**
   Ak text článku obsahuje aspoň dve vety s najmenej tromi slovami, sme schopný ho v tomto kroku algoritmu identifikovať. Môže nastať prípad, že ako jeden z nálezov bude identifikovaná časť prvého informačného odseku článku. Okrem časti popisu článku z RSS kanálu, ktorý slúži na jeho identifikáciu v zdrojovom kóde stránky, si pamätáme aj celý text popisu. Zistíme, či sa nájdený textový reťazec nachádza v tomto popise. Ak áno, spracovanie prejde na ďalší nález, pretože túto časť už sme identifikovali príslušným selektorom v predchádzajúcom kroku.

7. **Postupne prechádzaj všetky nálezy vyhovujúce maske. Ak je nájdený text súčasťou titulku, prvého informačného odseku článku, ktorý bol identifikovaný v predchádzajúcich krokoch, alebo bol vymazaný, prejdi na ďalší výskyt vyhovujúci maske. V opačnom prípade zisti, ktorý oddiel (`div` identifikovaný atribútom `class` alebo `id`) bezprostredne pokrýva celý tento text. Zapamätaj si tento oddiel spolu s informáciou o dĺžke textu, ktorý sa v ňom nachádza. Vymaž celý tento oddiel.**

8. **Z nájdených oddielov urči ten, ktorý v sebe obsahuje najdlhší text a skonštruuj z neho selektor podľa príslušných pravidiel.**

Tento krok vychádza z jednoduchého tvrdenia, že text článku je najdlhším súvislým textom na stránke.

Na výstupe dostaneme najmenej jeden a najviac dva selektory. Jeden v prípade, že prvý informačný odsek článku nie je zvlášť oddelený od ostatného textu. Dva selektory v prípade, že oddelený je.

6.2.3 Úspešnosť Algoritmus

Okrem algoritmu samotného je veľmi dôležité uviesť, na základe koľkých spravodajských serverov bol algoritmus vyvíjaný a sú nim úspešne podporované. To nám pomôže ukázať, že algoritmus funguje na dostatočne veľkej množine serverov, čím doložíme našu snahu o čo najväčšiu mieru univerzálnosti. Zoznam obsahuje tie najznámejšie a najnavštevovanejšie české spravodajské servery. Pretože takýchto spravodajských serverov je veľké množstvo, z časových dôvodov nemohla byť otestovaná a zabezpečená podpora všetky z nich.

Algoritmus bol vyvíjaný na množine 14-tich spravodajských serveroch, pričom sa podarilo dosiahnuť podporu všetkých z nich. Konkrétne sa jedná o tieto servery:

- idnes.cz
- ihned.cz
- lidovky.cz
- tn.nova.cz
- novinky.cz
- blesk.cz
- denik.cz
- sedmicka.cz
- e15.cz
- ceskenoviny.cz
- aktualne.centrum.cz
- tyden.cz
- euro.cz
- rozhlas.cz
- czechfreepress.cz

Podpora iných serverov nie je vylúčená.

6.3 Extrakcia textu článkov

V predchádzajúcej kapitole sme sa zaoberali algoritmom, ktorého úlohou je vyhľadať selektory pre daný RSS Kanál, podľa ktorých sa bude extrahovať text článkov. Poslednou časťou softvéru je veľmi jednoduchý programový modul, ktorý dokáže podľa selektorov vyextrahovať text článku. Vstupom je URL adresa zdroja (článku) a množina selektorov. Na výstupe dostávame vyextrahovaný text článku.
Je to záležitosť použitej programátorskej technológie, ktorej sa detailne venuje kapitola 7.2.

6.4 Analýza textu

Nachádzame sa v stave, kedy sme schopní získavať texty jednotlivých článkov zo zvolených informačných kanálov. Ak však chceme sledovať frekvenciu jednotlivých slov, je nevyhnutné s textom vykonať niekoľko dôležitých operácií, ktoré si stručne popíšeme.

6.4.1 Tokenizácia

Prvou základnou operáciu je *tokenizácia textu*. *Token* je ľubovoľná jednotka textu, ktorá rozširuje lingvistický význam pojmu slovo. Pri automatickej segmentácií textu sa za token považuje akýkoľvek reťazec znakov medzi dvoma „bielymi miestami“ (ang. „white-space“).

V našom prípade budú tokenmi jednotlivé slová zložené z písmen českej abecedy. Na ich dĺžke nezáleží. Môžu obsahovať aj číslovky, čo zabezpečuje, aby sme neprišli o dôležité tokeny ako napr. „win7“ alebo „čt24“. Frekvenciu samostatných čísel sledovať nebudeme, preto sú z tokenizácie automaticky vyradené.

6.4.2 Lematizácia a ekvivalentné slová

Lematizácia je proces, ktorý prevádza slová na základný gramatický tvar. Základným tvarom môže byť napr. nominatív jednotného čísla pri podstatných menách v češtine. Pri iných slovných druhoch, akými sú napr. slovesá, musíme myslieť na časovanie a pod. V počítačovom spracovaní textov sa jedná o opak derivácie (generovanie všetkých možných tvarov slova z jeho základnej slovníkovej podoby).

Realizuje sa príslušným nástrojom nazývaným lematizátor. Lematizátor môže pracovať na základe komplexnej vnútornej logiky, pomocou ktorej dokáže vytvoriť základný tvar slova. Vytvorenie takéhoto nástroja nie je jednoduchá záležitosť a vyžaduje si spoluprácu jazykovedcov. V našom systéme preto použijeme slovníkový lematizátor.
Jedná sa o jednoduchý nástroj, ktorý porovnáva slovo so slovami uloženými v slovníku ekvivalentných slov. Každý záznam sa skladá z koreňa slova a jeho podoby. Za tvorbu slovníku je zodpovedný správca systému, ktorý môže jednoducho spravovať skupiny ekvivalentných slov.

Ako príklad si uveďme tvary „slunci, sluncí, sluncím, slincích, sluncemi“. Všetko sú to

tvary slova „slunce“. Predstavme si, že v zozname štatistík máme pre každý tvar uvedenú samostatnú frekvenciu. V podstate ide o to isté slovo, preto chceme tieto tvary spolu s ich frekvenciami spojiť do jedného. Správca vloží tieto slovné tvary spolu s ich koreňom do slovníku ekvivalentných slov a o ostané sa už postará systém. Samozrejmosťou je spätná aktualizácia databázy doposiaľ nazhromaždďovaných slov.

6.4.3 Stop slová

Stop slová (ang. „stop words“) sú slová, ktoré nemajú samé o sebe žiaden vecný význam. V češtine ide predovšetkým o predložky, spojky, častice a niektoré ďalšie slová. Príkladom môžu byť spojky ako napr. „a“, „aby“ alebo predložky „na“, „pri“, „v“, atď. Sledovať frekvenciu týchto slov nemá žiaden význam, pretože sa nachádzajú takmer v každom článku a ich počet nepatrí medzi dôležite informácie, ktoré chceme získať.

Zoznam stop slov je uložený v databáze a užívateľ má možnosť ho dynamicky meniť podľa svojich predstáv. Ak sú stop slova modifikované počas fungovania aplikácie, samozrejmosťou je opäť následná aktualizácia databázy doposiaľ nazhromaždených slov. Každé slovo, ktoré sme označili ako stop slovo bude zo zoznamu odstránené. Čo sa týka jednoznakových stop slov, tie netreba ukladať do zoznamu stop slov, pretože sú zo spracovania vyraďované automaticky.

Pretože pri analýze textu článku je kontrola a vylučovanie stop slov poslednou časťou spracovania textu, je možné tzv. „lemovať na stop slová“. To znamená, že v rámci ekvivalentných slov môžme vytvárať ekvivalentné skupiny, ktorých koreňom je stop slovo a to bude odfiltrované pri kontrole stop slov. Všetko je v rukách správcu systému, ako sa rozhodne narábať s ekvivalentnými skupinami a stop slovami.

6.5 Návrh databázy

V tejto kapitole sa pozrieme ako bude vyzerá dátové úložisko pre naše dáta a dôležité entity evidované v našom systéme.

6.5.1 Funkčné požiadavky

V tejto časti sa budeme na systém pozerať len z pohľadu užívateľskej funkcionality. Na začiatku si uvedieme odpovede na najdôležitejšie otázky spojené s tvorbou systému, ktoré nám pomôžu pri špecifikácií funkčných požiadaviek a návrhu databázy.

Prečo nový systém?

Tento systém chceme preto, aby sme mohli monitorovať frekvenciu slov na českých spravodajských serveroch a mať k dispozícií štatistiky odzrkadľujúce dianie v spoločnosti.

Čomu má systém slúžiť?

Úlohou systému je zhromažďovať a ponúkať každodenné štatistiky o frekvenciách slov na českých spravodajských internetových serveroch.

Kto s ním bude pracovať?

Systém je určený najmä pre širokú verejnosť, preto ponúka jednoduché a zrozumiteľné užívateľské rozhranie, aby bol každý užívateľ schopný jednoducho si v ňom nájsť štatistické informácie, ktoré ho zaujímajú.
Okrem toho systém potrebuje mať svojho administrátora, ktorý bude zodpovedný za jeho správu.

Aké budú vstupy do systému?

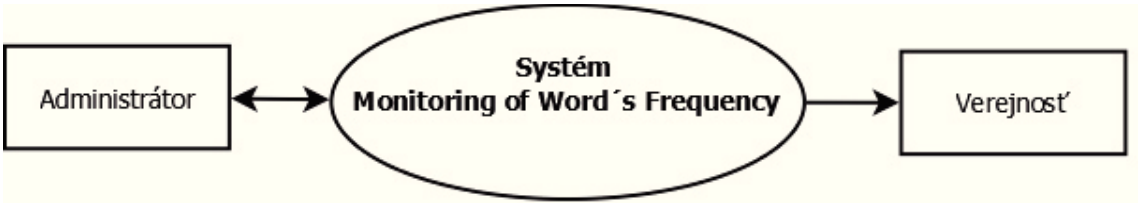
Pozn.: Tučným písmom sú zvýraznené jednotlivé **typy entít**, ktoré budú v systéme evidované, kurzívou *vlastnosti entít*. Pre lepšiu prehľadnosť sú uvedené v samotných zátvorkach.

Medzi informácie, ktoré majú byť v systéme evidované patria najmä jednotlivé RSS informačné kanály (**RSS kanály**). Každý RSS kanál je jednoznačne určený svojím identifikačným číslom (*ID RSS kanálu*). Ďalej nás zaujíma URL adresa RSS kanálu (*URL RSS kanálu*) a URL adresa serveru, ktorému kanál patrí, v úlohe názvu serveru (*Názov serveru*). Ďalej evidujeme selektory (*Selektory RSS kanálu*), podľa ktorých sa budú extrahovať texty jednotlivých článkov z RSS kanálu. Ako predposledné evidujeme v rámci RSS kanálu hash kód textového identifikátoru posledného prečítaného článku (*Hask kód posledného prečítaného článku*). Každý RSS kanál patrí do určitej kategórie resp. viacerých kategórií (*Kategória RSS kanálu*). Naopak jedna kategória môže obsahovať viacero RSS kanálov.

RSS kanál obsahuje skupinu článkov (**Články**). Pri článkoch nás zaujíma hash kód textového identifikátoru článku (*Hash kód článku*), identifikačné číslo RSS kanálu (*ID RSS kanálu*), z ktorého článok pochádza, titulok článku (*Titulok článku*), URL adresa článku (*URL adresa článku*), dátum zverejnenia (*Dátum zverejnenia*) a počet opätovných zverejnení v rodičovskom RSS kanále (*Počet zverejnení článku*).

Po spracovaní článkov dostávame k dispozícii jednotlivé evidované slová (**Slová**). Dôležité sú pre nás v spojitosti s RSS kanálom (*ID RSS kanálu*), z ktorého pochádzajú a dátumom zverejnenia článku (*Dátum zverejnenia*), ktorého sú súčasťou. Slová však nemajú žiadnu náväznosť na článok z ktorého pochádzajú, tj. nevieme, z ktorého článku slovo pochádza. Dôležitý údaj pri každom evidovanom slove je jeho číselná frekvencia za daný deň (*Frekvencia slova*).

Okrem zozbieraných slov evidujeme aj zoznam stop slov (**Stop slová**), ktoré poslúžia ako filter bezvýznamových slov a zoznam ekvivalentných slov (**Ekvivalentné slová**)



Obrázok 8: Kontextový diagram

vo forme slovo a jeho koreň, ktoré poslúžia ako lematizačný filter. Ako posledné uchovávame zoznam administrátorov (**Administrátori**), kde každý z nich má svoje užívateľské meno (*Meno administrátora*) a prístupové heslo (*Heslo administrátora*).

Aké budú výstupy so systému?

Primárnymi výstupmi zo systému budú frekvenčné štatistiky slov a článkov na spravodajských serveroch. Tieto štatistiky má užívateľ možnosť zobraziť si podľa spravodajského serveru, podľa kategórie a podľa časového obdobia takto:

- pre každý deň zvlášť,
- za posledný týždeň,
- za posledný mesiac,
- za posledný rok,
- alebo zadať špecifický časový interval.

Sekundárne bude výstupom zoznam evidovaných RSS kanálov, stop slov a ekvivalentných slov.

Aké funkcie bude systém plniť?

V rámci RSS kanálov systém umožňuje ich pridávanie a mazanie. So zmazaním RSS kanálu sa automaticky mažú aj všetky slová a články získané z tohto kanálu. Vkladanie slov je automatickou záležitosťou, o ktorú sa stará systém. Ďalej umožňuje pridávanie, editáciu a mazanie stop-slov a ekvivalentných slov. Podrobný zoznám funkcií systému je uvedený v tabuľke udalostí a reakcií systému 3.

Aké je okolie systému?

Ako je vidieť z kontextového diagramu na obrázku 8, okolie systému je tvorené administrátorom a verejnosťou. Verejný užívateľ, na rozdiel od administrátora, nemôže nijakým spôsobom zmeniť vnútorný stav systému, alebo jeho dátovú vrstvu

6.5.2 Dátová analýza - Konceptuálny dátový model riešenia

Na začiatku dátovej analýzy si uvedieme lineárny textový zápis našej databázy, vyplývajúci z textového popisu vyššie uvedených funkčných požiadaviek. Zápis začína uvedením **typu entity** tučným písmom, za ktorým v zátvorkách nasledujú jeho vlastnosti. Vlastný kľúč je označený podčiarknutím, *cudzí kľúč* kurzívou, na označenie vzťahov medzi entitnými typmi použijeme KAPITÁLKY.

Pre pomenovanie vlastností je zvolená maska, podľa ktorej názvu samotnej vlastnosti vždy predchádza názov entitného typu oddelený podčiarkovacím znamienkom. Toto značenie je zvolené zámerné, aby bolo pri pohľade na hociktorú vlastnosť ihneď zrejmé, ktorému entitnému typu náleží, čo prináša vyššiu mieru jednoznačnosti do návrhu databázy. Podčiarkovacie znamienko je okrem toho použité aj na oddelenie viac slovných pomenovaní atribútov. *Už v tejto časti návrhu sme automaticky rozložili vzťah M:N medzi RSS kanálmi a kategóriami pomocou väzobnej tabuľky.* Na lineárny zápis sa preto môžeme pozerať ako na zápis výsledný. Pri návrhu bol použitý anglický jazyk.

- **rss_channels** (rss_channels_id, rss_channels_channel, rss_channels_server_url, rss_channels_selectors, rss_channels_last_read_article_ghash)
- **articles** (articles_guide_hash, *rss_channels_id*, articles_title, articles_link, articles_date, articles_count)
- **words** (words_word, *rss_channels_id*,words_date, words_count)
- **equivalent_words** (equivalent_words_word, equivalent_words_root)
- **stop_words** (stop_words_word)
- **rss_channels_categories** (*rss_channels_id*, *categories_category*)
- **categories** (categories_category)
- **users** (users_name, users_password)

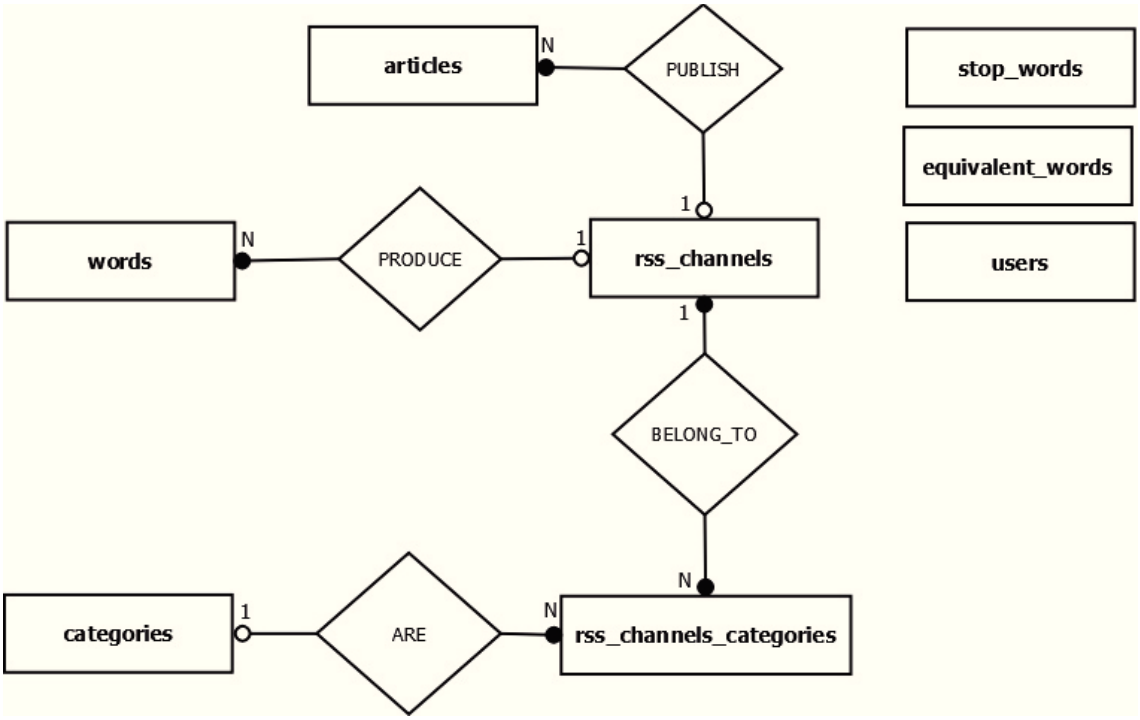
Vzťahy, kardinalita, povinnosť členstva

Jeden RSS kanál uverejňuje viacero článkov. Jeden článok je súčasťou jedného RSS kanálu (1:N). Každý článok musí mať svoj RSS kanál. RSS kanál nemusí mať žiaden článok.

Jeden RSS kanál obsahuje viacero slov za daný deň. Jedno slovo za daný deň je vždy pre jeden RSS kanál(1:N). Každé slovo musí mať svoj RSS kanál. RSS kanál nemusí mať žiadne slová.

Jeden RSS kanál môže patriť do viacerých kategórií. Jedna kategória môže obsahovať viacero RSS kanálov (M:N). Každý RSS kanál musí patriť do nejakej kategórie. Kategória nemusí obsahovať žiadne RSS kanály.

Z popisu získavame medzi niektorými typmi entít nasledujúce typy vzťahov:



Obrázok 9: Entity Relationship Diagram

- PUBLISH(**rss_channels**, **articles**)
- PRODUCE(**rss_channels**, **words**)
- BELONG_TO(**rss_channels**, **rss_channels_categories**)
- ARE(**categories**, **rss_channels_categories**)

Konceptuálny model databázy doplníme graficky pomocou ER Diagramu (9).

Dátový slovník

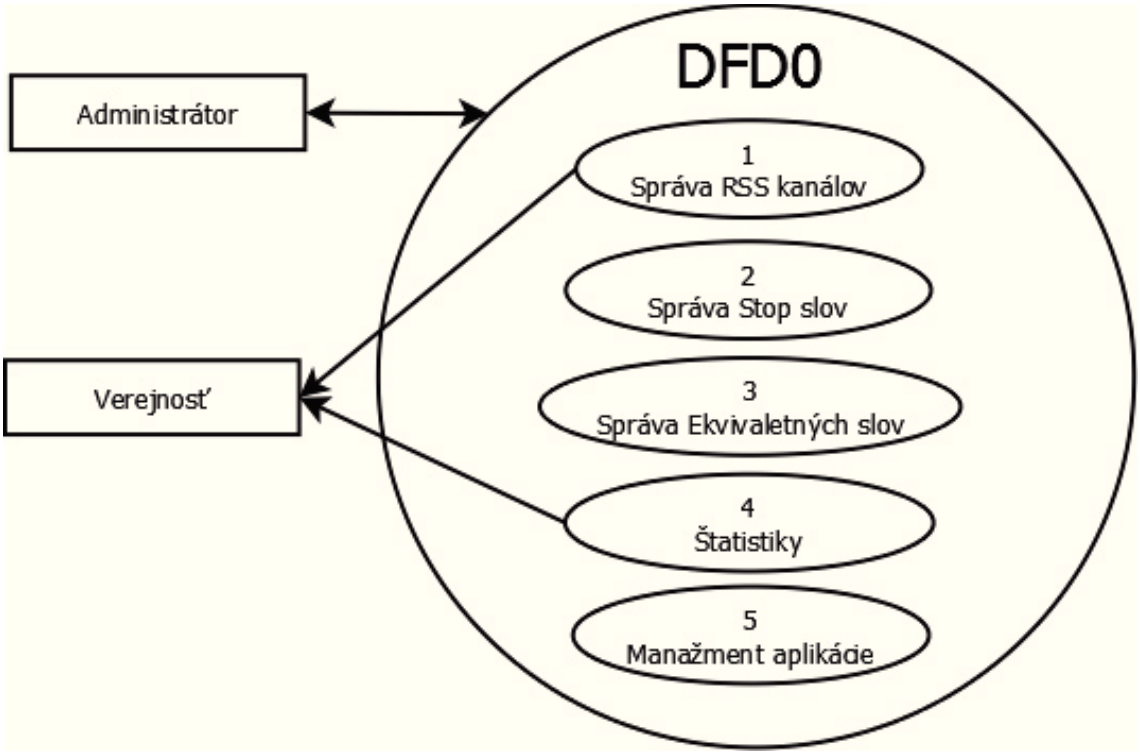
Dátový slovník je súčasťou prílohy A.

6.5.3 Funkčná analýza

Funkčný model systému zachytíme pomocou hierarchickej štruktúry DFD.

Kontextový diagram

Na vrchole hierarchie stojí kontextový diagram uvedený na obrázku 8. Obsahuje celý systém ako jednu funkciu, definuje hranice systému a všetkých aktérov (zdroje a ciele



Obrázok 10: DFD úroveň 0

dát). Systém na tejto úrovni chápeme ako čiernu skrinku s definovanými vstupmi a výstupmi.

DFD úrovne 0

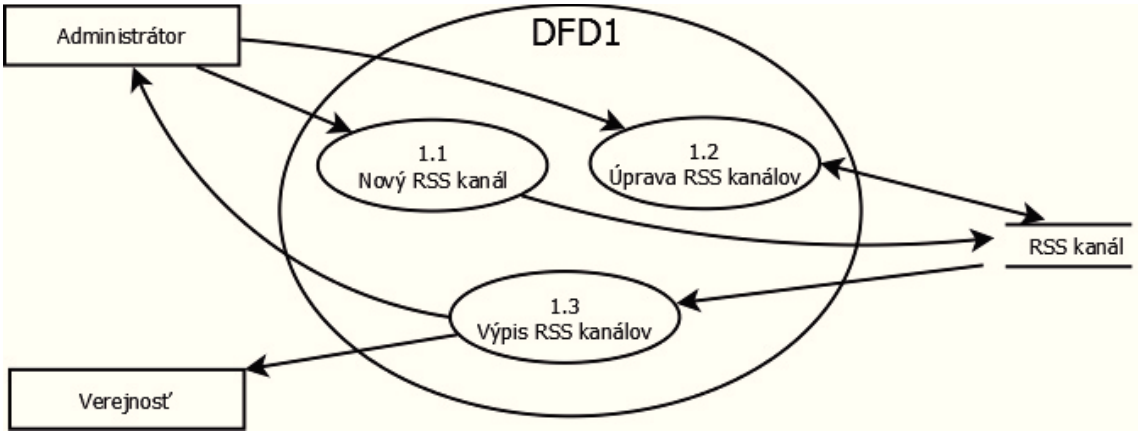
Bezprostredným rozkladom kontextového diagramu je DFD úrovne 0 uvedený na obrázku 10. Obsahuje základné funkcie systému (rozklad na subsytémy).

DFD úrovne 1

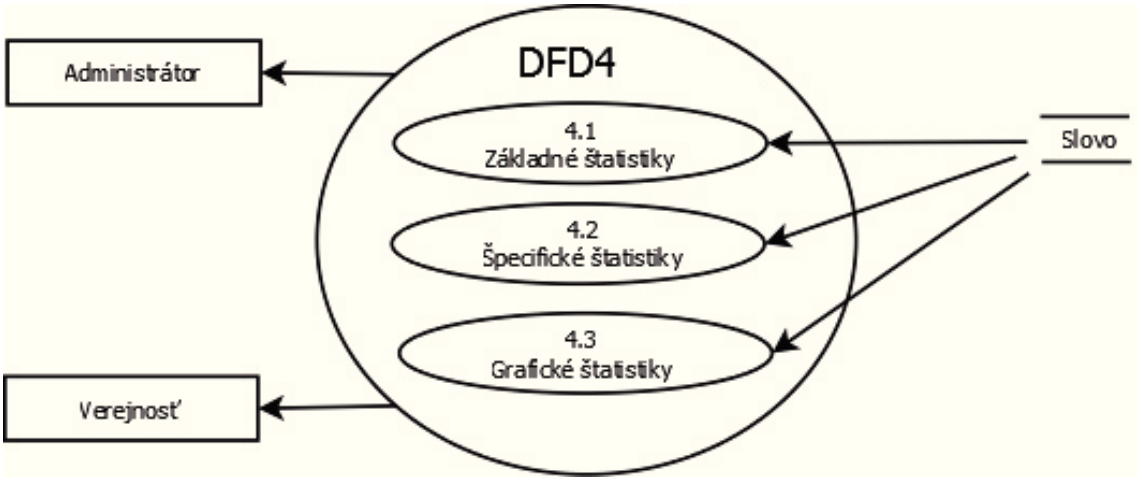
Správu RSS kanálov môžeme ďalej rozložiť na elementárnu úroveň užívateľsky ďalej nedeliteľných funkcií (obr. 11). Rozklad funkcií 2 a 3 z DFD diagramu úrovne 0 je obdobný.

DFD úrovne 4

Užívateľ má možnosť zobraziť si niektorú z troch druhov štatistík (obr. 12). Základné štatistiky poskytujú zobrazenie frekvencií slov alebo článkov pre všetky servery a slova dohromady a časovo pre každý deň zvlášť. Špecifické štatistiky umožňujú špecifikovať výsledný frekvenčný zoznam, podľa serveru, kategórie a časového obdobia. Grafické



Obrázok 11: DFD úroveň 1



Obrázok 12: DFD úroveň 4

štatistiky umožňujú zobraziť frekvencie slov pomocou histogramu, kde je jasne vidieť frekvencie za jednotlivé dni.

Udalosť	**Reakcia Systému**	**Aktér**
Nový RSS kanál	Zapíš do zoznamu RSS kanálov.	Administrátor
Nové stop slovo	Zapíš do zoznamu stop slov.	Administrátor
Nové ekvivalentné slovo	Zapíš do zoznamu ekvivalentných slov.	Administrátor
Uprav stop slovo	Uprav informácie v zozname stop slov.	Administrátor
Uprav ekvivalentné slovo	Uprav informácie v zozname ekvivalentných slov.	Administrátor
Zmaž RSS kanál	Zmaž zvolené záznamy zo zoznamu RSS kanálov a zároveň zmaž príslušné slová zo zoznamu slov.	Administrátor
Zmaž stop slovo	Zmaž zvolené záznamy zo zoznamu stop slov.	Administrátor
Zmaž ekvivalentné slovo	Zmaž zvolené záznamy zo zoznamu ekvivalentných slov.	Administrátor
RSS kanály	Vypíš zoznam RSS kanálov.	Administrátor, Verejnosť
Stop slová	Vypíš zoznam stop slov.	Administrátor
Ekvivalentné slová	Vypíš zoznam ekvivalentných slov.	Administrátor
Základné štatistiky	Vypíš zoznam štatistík.	Administrátor, Verejnosť
Špecifické štatistiky	Špecifikuj zoznam štatistík.	Administrátor, Verejnosť
Grafické štatistiky	Zobraz grafické štatistiky pre slovo.	Administrátor, Verejnosť
Štart monitorovania	Spusť sledovanie frekvencie slov .	Administrátor
Stop monitorovania	Zastav sledovanie frekvencie slov .	Administrátor
Informácie	Zobraz informácie o aplikácií.	Administrátor, Verejnosť
Logy	Zobraz informačné logy aplikácie.	Administrátor

Tabulka 3: Tabuľka udalostí a reakcií v systéme

7 Implementácia

Pre implementáciu programu bol zvolený môj obľúbený objektovo orientovaný programovací jazyk Java. Projekt bol vyvíjaný v integrovanom vývojovom prostredí Eclipse Java EE IDE for Web Developers verzie Indigo Service Release 2. Jazyk Java ponúka okrem svojho jadra množstvo zaujímavých knižníc použiteľných pri riešení čiastkových úloh v tomto systéme.

7.1 Java Informa

Na parsovanie RSS kanálov bola použitá knižnica Java Informa[11], ktorá ponúka pohodlné aplikačné rozhranie pre prácu s RSS kanálmi. Informa je šírená pod licenciou „GNU Library“ alebo „Lesser Public Licence“ a je zadarmo. Umožňuje univerzálny prístup nezávislý na verzii RSS kanálu.

Cieľom Informa projektu je poskytnúť novú agregačnú knižnicu založenú na platforme Java a unifikovať všeobecné požiadavky do knižnice, ktorá môže byť použitá každým vývojárom, ktorý potrebuje pracovať s RSS vo svojom programe. Snaží sa o harmonizovaný pohľad na nový objektový model RSS. *Použitá bola aktuálna verzia 0.7.0 (Alpha 2).*

Knižnica umožňuje publikovať, čítať, zisťovať zmeny a filtrovať RSS kanály. Najdôležitejšou funkcionalitou, ktorú využijeme, je čítanie RSS kanálu a jeho jednotlivých položiek (článkov). Pozrime sa preto, ako je možné pomocou knižnice Java Informa získavať položky z RSS kanálu a mať ich k dispozícii pre ďalšie spracovanie.

Aby bolo možné prečítať RSS kanál, potrebujeme zistiť jeho URL adresu (z internetu), aby sme ho mohli začať analyzovať (*ang. parse*).
Informa na účely parsovania ponúka triedu `de.nava.informa.parsers.FeedParser`, ktorá má sadu statických analytických metód `static parse()`. Tieto metódy požadujú dva typy parametrov:

- `de.nava.informa.core.ChannelBuilderIF`, ktorý sa používa na vytvorenie objektového modelu RSS kanálu. Máme na výber medzi úložiskom v pamäti počítača a perzistentným databázovým úložiskom.
- RSS Dokument, ktorý ma byť analyzovaný. Môže byť zadaný vo forme URL, ako súbor, alebo ako objekt typu `Reader`, `InputStream` alebo `InputSource`.

Následuje konkrétny príklad vytvorenia objektového modelu RSS kanálu pomocou knižnice Informa pre RSS kanál kategórie „Šport“ zo serveru *IDnes.cz*.

```
URL url = new URL( "http://servis.idnes.cz/rss.aspx?c=sport" );
ChannelIF channel = FeedParser.parse( new ChannelBuilder(), url );
```

Výpis 4: *Java Informa*: Vytvorenie objektového modelu RSS kanálu

Používať budeme `de.nava.informa.core.ChannelBuilder` prvú implementáciu `ChannelBuilderIF` rozhrania. Toto rozhranie implementuje ešte jednu triedu.
Je ňou `de.nava.informa.impl.hibernate.ChannelBuilder`. Rozdiel medzi nimi je ten, že prvá z nich prečíta RSS kanál a uloží ho do pamäti počítača, zatiaľ čo druhá ho ukladá do databázy nakonfigurovanej pre perzistenciu pomocou Hibernate. Pre naše potreby je vhodnejšia jednoduhšia implementácia, ktorá si ukladá dáta do pamäte počítača.

Rozhranie `ChannelIF` poskytuje informácie o kanále samotnom a takisto aj prístup k jednotlivým položkám v tomto kanále. Pre kanál môžme získať tieto informácie:

- titulok/názov kanálu,
- popis kanálu,
- obrázok reprezentujúci tento kanál,
- formát kanálu,
- dátum publikácie,
- dátum poslednej aktualizácie,
- vydavateľ kanálu,
- atď.

Nie každá verzia RSS špecifikácie ponúka všetky tieto informácie, preto je najskôr potrebné skontrolovať, ktoré z nich RSS kanál obsahuje, pred tým ako s nimi začneme pracovať. Takto môžme získať titulok a popis samotného kanálu, ako je vidieť na nasledujúcom príklade:

```
System.out.println( "Channel=" + channel.getTitle() );
System.out.println( "Description=" + channel.getDescription() );
```

Výpis 5: *Java Informa*: Získavanie informácií o RSS kanále

Všeobecné informácie o kanále samotnom nie sú také dôležité, ako informácie o samotných položkách, ktoré obsahuje. Získať ich môžme pomocou metódy `getItems()`, ktorá vracia objekt typu `java.util.Collection` skladajúci sa z prvkov typu `de.nava.informa.core.ItemIF`. Objekty typu `ItemIF` ďalej poskytujú konkrétne informácie o jednotlivých článkoch. Ako sme už v kapitole 3 uviedli, využívať budeme hlavne titulok a popis článku, dátum publikácie, URL adresu a jednoznačný textový identifikátor článku.

Príklad praktickej programátorskej práce s článkami v RSS kanále môže vyzerať nasledovne:

```
Collection items = channel.getItems();
for( Iterator i=items.iterator(); i.hasNext(); )
{
   ItemIF item = ( ItemIF )i.next();

   System.out.println( "Title=" + item.getTitle() );
   System.out.println( "Description=" + item.getDescription() );
   System.out.println( "Date=" + item.getDate() );
   System.out.println( "Link=" + item.getLink() );
   System.out.println( "Guid=" + rssItem.getGuid().getLocation() + "\n" );
}
```

Výpis 6: *Java Informa*: Práca s jednotlivými článkami v RSS

Pomocou metódy `getItems()` získame z objekt typu `Channel` kolekciu všetkých položiek RSS kanálu. Jednotlivé položky prechádzame v cykle `for` a pomocou príslušných metód dostávame informácie o nich.

7.2 JSoup

JSoup [12] je Java knižnica pre prácu s reálnymi HTML zdrojmi. Jedná sa o projekt s otvoreným zdrojovým kódom (*ang. open source project*) distribuovaný pod MIT licenciou. Ponúka veľmi pohodlné API pre extrakciu a manipulovanie s dátami s použitím toho najlepšieho z DOM, CSS a JQuery podobných metód.

JSoup implementuje WHATWG HTML5 špecifikáciu a parsuje HTML do rovnakého DOM ako to robia moderné prehliadače. JSoup je navrhnutý tak, aby si poradil so všetkými druhmi HTML formátu od najstarších až po najnovšie. Použitá bola verzia 1.6.2. Medzi najväčšie výhody patrí:

- Získavanie a analýza HTML priamo z URL, súboru alebo textového reťazca.
- Hľadanie a extrakcia dát s využitím analýzy dokumentového objektového modelu alebo CSS selektorov.
- Manipulácia s HTML elementmi, atribútmi a textom.
- Výstupom je „upratený“ HTML kód.

Pri parsovaní HTML dokumentu sa parser za každú cenu snaží o čo najpresnejšiu analýzu HTML zdroja, ktorý sme mu poskytli, bez ohľadu na to, či sa jedná o dobre formátovaný HTML kód alebo nie. Dokáže si poradiť s:

- neuzatvorenými HTML značkami (napr. `<p>Dobrý <p>deň` sparsuje ako `<p>Dobrý</p> <p>deň</p>`),
- implicitnými značkami (napr. voľné `<td>Tabuľkové dáta</td>` je zabalené do `<table><tr><td>Tabuľkové dáta</td></tr></table>`),

Parser spoľahlivo vytvára HTML dokumentovú štruktúru obsahujúcu základné časti `head` a `body` a iba vhodné elementy v rámci nich.

V kapitole 6.2 sme sa zaoberali vyhľadávaním selektorov, ktoré určujú, kde sa nachádza text článku na danej stránke a serveri. Knižnica JSoup je v tomto smere pre nás veľmi dôležitá, pretože podporuje syntax podobný CSS (alebo jQuery) selektorom na vyhľadávanie elementov, ktoré umožňujú veľmi účinné a robustné dotazy. Podľa týchto selektorov dokáže vyextrahovať text článku z HTML stránky. Konkrétne sa jedná o metódu `select(String selector)`, ktorá je k dispozícií pre JSoup objekty typu `Document`, `Element` alebo `Elements`. Metóda vracia zoznam elementov, ktoré poskytujú množstvo metód pre extrakciu a manipulovanie s výsledkom. Syntax a sémantika selektorov, ktorú budeme využívať ukazuje tabuľka 4. Takto sme sa dopracovali k výslednej podobe selek-

Vzor	**Výsledné nálezy**	**Príklad**
`tag`	Všetky elementy s daným názvom	`div`
`[attr=val]`	Všetky elementy ktorých atribút s názvom `attr` obsahuje hodnotu `val`	`[class=opener]`
`tag[attr=val]`	Všetky elementy `tag`, ktorých atribút `attr` obsahuje hodnotu `val`	`div[class=opener]`

Tabulka 4: JSoup selektory - syntax a sémantika

torov, ktoré budeme používať a ktoré nám umožnia vybrať z HTML dokumentu len tie elementy, v ktorých sa skrýva text článku. Poslednou časťou je získať text z vybraných elementov. K tomu poslúži metóda `Element.text()`, ktorá vracia textový obsah daného elementu. V prípade, že obsahom elementu sú ďalšie HTML značky (typickým príkladom je text článku rozdelený na textové odseky pomocou párovej značky
`<p>textový odsek</p>`), je aj z nich vyextrahovaný textový obsah. Vo všeobecnosti môžme hovoriť, že podľa daného selektoru bude vyextrahovaný celý textový obsah vybraných elementov bez ohľadu na to, či tie ďalej obsahujú vo svojom vnútri ďalšie elementy.
Ďalšou výhodou je, že ak vyextrahovaný text obsahuje špeciálne znakové HTML entity ako napr. ` ` zastupujúcú pevnú, nedeliteľnú medzeru a ďalšie, tie sú automaticky prevedené do svojej primárnej podoby. Rovnako pracuje s takýmito špeciálnymi značkami aj webový prehliadač.

V tejto chvíli už dokážeme vyextrahovať text článku podľa selektorov. Ďalej nasleduje záverečná tokenizácia textu (rozdelenie na jednotlivé slová) a ukladanie slov do databázy.

7.3 SRBD

Ako SRBD pre ukladanie dát bol použitý Microsoft SQL Server 2008 R2. Pre implementáciu bol použitý originálny JDBC ovládač poskytovaný výrobcom SRBD (mysql-connector-java-5.1.17).

V jazyku T-SQL na strane SRBD bola vytvorená procedúra, ktorá sa stará o spätnú lematizáciu všetkých evidovaných slov podľa novej skupiny ekvivalentných slov.

V prílohe v kapitole B sú uvedené ukážkové dopyty používané pre výpis najrozličnejších kombinácií špecifických štatistík.

7.4 Webová aplikácia - Monitoring of Word's Frequency

Výsledná podoba systému je požadovaná vo forme webovej aplikácie. Na jej tvorbu bol použitý java-webový aplikačný framework JSF 2.0., ktorý sa teší mojej obľube. Cieľom tohto rámca je zjednodušiť vývoj webových užívateľských rozhraní. Je to webový rámec na tvorbu užívateľských rozhraní pomocou komponent. Zachováva návrhový vzor MVC. JSF 2.0 používa ako zobrazovaciu technológiu Facelety na rozdiel od verzie 1.x, ktorá používala JSP stránky.

Pri tvorbe aplikácie pomocou základných užívateľských JSF komponent boli navyše použité pokročilejšie komponenty PrimeFaces, ako napr. kalendáre alebo histogram na zobrazovanie grafických štatistík.

Webová aplikácia je nasadená a prístupná v rámci školskej siete na servery dbedu.cs.vsb.cz, konkrétne na adrese:

http://dbedu.cs.vsb.cz:9090/CIN020_NewsMonitoring

Prístup do administrátorského režimu je možný po prihlásení s použitím prihlasovacích údajov:

- Meno: **admin**
- Heslo: **fei2012**

7.5 Webový server Apache Tomcat

Aby mohla webová aplikácia vytvorená v JSF 2.0 fungovať, je potrebné nasadiť ju na webový server umožňujúci jej beh. Apache Tomcat je webový kontajner (server) s otvoreným zdrojovým kódom implementujúci Java Servlet a Java Server Page technológie. Dôvodom pre jeho voľbu boli predchádzajúce skúsenosti s nim. *Použitá bola verzia 7.0.11.*

7.6 Systémové príručky

Súčasťou prílohy je kapitola C, ktorá obsahuje veľmi podrobný popis celej webovej aplikácie, ktorý sa dá použiť aj ako užívateľský manuál. Okrem toho je možné v ňom nájsť detailný popis celého výsledného užívateľského rozhrania.

Programátorská príručka vygenerovaná pomocou nástroju JavaDoc je súčasťou priloženého CD.

8 Testy a štatistiky

V tejto kapitole budeme testovať niektoré funkcie výsledného systému a takisto sa pozrieme na zaujímavosti v zozbieraných štatistikách. Tým ukážeme, že systém splňuje naše očakávania a dokáže zachytiť dôležité udalosti v spoločnosti. Zaoberať sa budeme len základnými štatistikami, aby sme demonštrovali správnu funkčnosť systému. Špecifické štatistiky je možné zobraziť si priamo v systéme.

Do monitorovania bola vybraná a priradená testovacia množina 32 RSS kanálov, pre 6 rôznych serverov a pre rozličné kategórie. Ich zoznam, spolu s monitorovanými kategóriami (originálne názvy z aplikácie) sa nachádza v tabuľke 5.
Do sledovania mohlo byť samozrejme pridelené viacero zdrojov, ale pre účely testovania a získania ukážkových štatistík je dostačujúca aj uvedená množina.
Naším cieľom v tomto kroku preto nebolo dosiahnuť kvantitu nazbieraných dát, ale demonštrovať kvalitu softvéru.

8.1 Pridávanie nových RSS kanálov

Na začiatku sa zameriame na pridávanie nových RSS kanálov do spracovania. Konkrétne sa budeme zaujímať o čas potrebný na to, aby algoritmus vyhľadal selektory pre vybrané RSS kanály. Pre každý z uvedených testovacích serverov bude vybraný jeden vzorový RSS kanál. Testovať všetky je zbytočné, pretože v rámci jedného serveru majú RSS kanály rovnakú formu (tj. najmä počet článkov v nich).

Výsledky sú prezentované vo forme tabuľky 6 , kde bude uvedené, z ktorého serveru kanál pochádza. Kategória nie je v tomto prípade dôležitá Ďalej nás bude zaujímať počet článkov v RSS ,čas potrebný na určenie selektorov (v tabuľke ako Čas 1) a čas potrebný na stiahnutie zdrojových kódov všetkých článkov z kanálu (v tabuľke ako Čas 2).
Z tabuľky vidno, že rozdiely sú ako v celkovom čase potrebnom na vyhľadanie selektorov, tak aj v čiastkovom čase potrebnom na stiahnutie zdrojových kódov.

8.2 Najčastejšie slová

Základnou štatistickou informáciu je zoznam najčastejšie používaných slov v článkoch. V tabuľke 7 je uvedený rebríček desiatich najčastejších slov za časové obdobie od 14.5.2012 do 10.6.2012. Slová sú pre všetky servery a kategórie dohromady.

8.3 Najdôležitejšie udalosti

Zhromažďovanie článkov spolu s indikátorom ich dôležitosti nám podáva ucelený obraz o najdôležitejších udalostiach v zvolenom časovom období. Za časové obdobie monitorovania od 14.5.2012 do 10.6.2012 sa na popredných pozíciách najdôležitejších udalosti umiestnili články uvedené v tabuľke 8 (10 najdôležitejších). Ďalšie články pre konkrétne kategórie je možné dohľadať priamo v aplikácií.

Server	Kategórie
http://www.idnes.cz/	Praha, Brno, Ostrava, news, football, ice hockey, tennis, volleyball, basketball, sport, econimics, culture, mobil, auto, xman, she.
http://ihned.cz/	politics, czech, global, news, technic, art, life.
http://www.lidovky.cz/	czech, home, global, sport, culture, science, auto, invisible dog.
http://tn.nova.cz/	news, tn.cz.
http://novinky.cz/	news.
http://www.blesk.cz/	news.

Tabulka 5: Testovacia množina RSS kanálov.

Server	Počet článkov v RSS	Úspešnosť	Čas1	Čas2
http://www.idnes.cz/	25	92%	17 s	4 s
http://ihned.cz/	30	86%	75 s	11 s
http://www.lidovky.cz/	30	93%	12 s	3 s
http://tn.nova.cz/	30	80%	26 s	9 s
http://novinky.cz/	25	64%	29 s	7 s
http://www.blesk.cz/	25	72%	55 s	22 s

Tabulka 6: Čas potrebný na pridávanie nových kanálov.

8.4 Grafické zobrazenie frekvenice pre slová

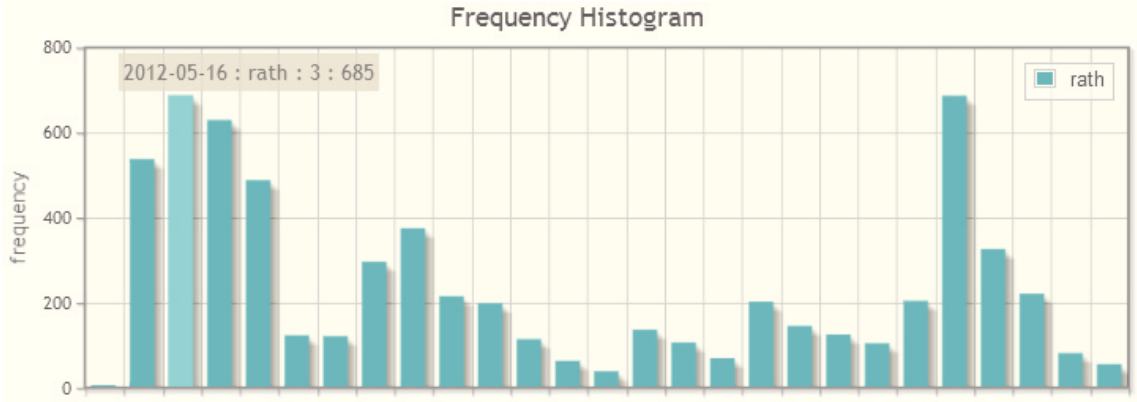
Veľmi zaujímavé informácie môžu byť získané pri zobrazení histogramu pre niektoré slová. Ako ukážku si môžeme uviesť histogram pre slovo „rath“ (obr. 13). V grafe si môžeme všimnúť vyznačeného stĺpca s maximálnou frekvenciou pre deň 15.5.2012. Toto maximum môžeme dať do súvislosti s článkom umiestneným na druhom mieste v najdôležitejších udalostiach s dátumom 15.5.2012 a vidíme tak, že najčastejšie slová a najdôležitejšie udalosti spolu úzko súvisia.

Pozícia	Slovo
1	rok
2	rath
3	policie
4	let
5	první
6	koruna
7	člověk
8	foto
9	david
10	mluvčí

Tabulka 7: Najčastejšie slová za obdobie od 14.5.2012 do 10.6.2012.

Článok	Dátum
U Bzence hoří rozlehlý les. Hasí ho i helikoptery	24.5.2012
Policejní komando zatklo hejtmana Davida Ratha (ČSSD)	15.5.2012
Na tuningovém srazu vlétla dvě auta přímo do diváků!	19.5.2012
Silné zemětřesení opět zasáhlo Itálii. Praskala zem, padaly budovy	29.5.2012
Rath odjel z věznice do sněmovny. Miliony v krabici bude vysvětlovat poslancům	22.5.2012
Sever Itálie zasáhlo další zemětřesení, stejně silné jako před týdnem	29.5.2012
Dagmar (†31) zabil na Novém Zélandě sexuální deviant	29.5.2012
Ptejte se vyřazených z Hlasu ČeskoSlovenska na vše, co vás zajímá	28.5.2012
Břeclav den po přiznání: Nervózní jsou obyvatelé i starosta	24.5.2012
Rathova zástupce Šejnostu zmlátili	22.5.2012

Tabulka 8: Najdôležitejšie udalosti za obdobie od 14.5.2012 do 10.6.2012.



Obrázok 13: Frekvenčný histogram pre slov „rath“.

9 Záver

Zaujímavými výsledkami experimentálnych testov sa podarilo preukázať splnenie hlavnej úlohy tejto práce, ktorou bolo sledovanie frekvencie slov na českých internetových spravodajských serveroch. Bolo dokázané, že program skutočne dokáže zachytiť „ducha doby“ v podobe toho najdôležitejšieho dianie v spoločnosti, čo bolo naším cieľom.

Zdrojom pre články sa stali RSS kanály, ktoré boli monitorované v pravidelných časových intervaloch z dôvodu sledovania nových článkov.
Bol vytvorený komplexný algoritmus, ktorý dokáže určiť, kde presne na webových stránkach reprezentujúcich články z konkrétneho RSS kanálu sa nachádza text článku. Výsledkom tohto algoritmu je množina špeciálnych selektorov, ktoré slúžia ako vstup pre nástroj na extrakciu textu z článkov.
Extrahovaný text je ďalej rozdelený na jednotlivé slová, ktoré sú filtrované pomocou zoznamu stop slov a upravované na základný tvar pomocou slovníku ekvivalentných slov.

Okrem pôvodného zadania zadania zahrňujúceho sledovanie frekvencie slov sa podarilo rozšíriť softvér aj o sledovanie článkov samotných a vyhľadávanie v nich. Výstupom je rebríček najdôležitejších článkov za ľubovoľné časové obdobie, ktoré zachycujú dôležité udalosti v spoločnosti. Okrem toho je možné zobrazovať si články z ľubovoľných kategórií pre ľubovoľné časové obdobie napr. aj konkrétne dni, čo sa ukázalo ako veľmi zaujímavá a prakticky dobre využiteľná funkcionalita.

Ako autor projektu si dovoľujem tvrdiť, že prínos tejto práce je celkom významný, pretože sa jedná o poskytovanie unikátnych informácií v rámci internetovej žurnalistiky v Českej republike. Výsledné zoznamy sú určené pre každého záujemcu, ktorý sa bude chcieť pozrieť na konkrétne štatistiky, alebo si vyhľadať články podľa svojho záujmu.

Čo sa týka rozšírenia aplikácie, v budúcnosti by mohla by byť doplnená o možnosť registrácie verejných užívateľov. Tým by si mohol každý na svojom privátnom účte pridávať informačné kanály podľa toho, čo ho zaujíma a sledovať tak nielen frekvencie slov, ale aj článkov zo svojich obľúbených zdrojov.

10 Reference

Internet

[1] AMBROŽ, Jan. *Internet versus tisk – co (ne)mají společného?* Lupa.cz - server o českém Internetu [online]. 3.4.2008 [cit. 2012-05-15]. Dostupné z: http://www.lupa.cz/clanky/internet-versus-tisk-co-nemaji-spolecneho/

[2] NEFF, Ondrej. *NEVIDITELNÝ PES: První český ryze internetový deník.* [online]. 23.4.1996 [cit. 2012-05-15]. Dostupné z: http://neviditelnypes.lidovky.cz/

[3] GOOGLE. *Google Trends* [online]. ©2008 [cit. 2012-05-15]. Dostupné z: http://www.google.com/trends/

[4] GOOGLE. *Google Insights for Search* [online]. ©2012 [cit. 2012-05-15]. Dostupné z: http://www.google.com/insights/search/

[5] WIKIMEDIA FOUNDATION, INC. *Wiktionary: Frequency lists - Wiktionary* [online]. ©2006 [cit. 2012-05-15]. Dostupné z: http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists

[6] *Český národní korpus: Srovnávací frekvenční seznamy* [online]. Ústav Českého národního korpusu FF UK, Praha 2010. Dostupné z WWW: http://ucnk.ff.cuni.cz/srovnani10.php.

[7] OPENSUBTITLES.ORG. *Titulky - sťahujte titulky pre filmy DivX z najväčšej otvorenej titulkovej databázy.* [online]. [cit. 2012-06-10]. Dostupné z: http://www.opensubtitles.org/

[8] BERKMAN CENTER. *RSS 2.0 Specification (RSS 2.0 at Harvard Law)* [online]. ©2003 [cit. 2012-05-23]. Dostupné z: http://cyber.law.harvard.edu/rss/rss.html

[9] BUREŠ, Jirí. *RSS 2.0 — Interval.cz* [online]. 16.09.2004 [cit. 2012-05-15]. Dostupné z: http://interval.cz/clanky/rss-20/

[10] THE LIBRARY OF CONGRESS. *SO 639-2 Registration Authority - Library of Congress* [online]. 18.10.2010 [cit. 2012-06-06]. Dostupné z: http://www.loc.gov/standards/iso639-2/

[11] SCHMUCK, Niko. *Informa: RSS Library for Java - Overview* [online]. ©2002-2007 [cit. 2012-05-16]. Dostupné z: http://informa.sourceforge.net/

[12] HEDLEY, Jonathan. *JSoup: Java HTML Parser, with best of DOM, CSS, and jquery* [online]. ©2009-2012 [cit. 2012-05-22]. Dostupné z: http://jsoup.org/

[13] W3C. *Tim Berners-Lee* [online]. 2012. Dostupný z WWW: <http://www.w3.org/People/Berners-Lee/>.

[14] W3C. *W3C : Worl Wide Web Consortium* [online]. 2012. Dostupný z WWW: <http://www.w3.org/>.

[15] W3C. *HTML Current Status - W3C* [online]. 2012 [cit. 2012-06-10]. Dostupné z: http://www.w3.org/standards/techs/html#w3c_all

[16] JANOVSKÝ, Dušan. *Jak psát web, návod na html stránky* [online]. 16.5.2012 [cit. 2012-06-06]. Dostupné z: http://www.jakpsatweb.cz/

[17] SPIR Z. S. P. O. *NetMonitor* [online]. ©2011 [cit. 2012-06-06]. Dostupné z: http://www.netmonitor.cz/

A Dátový slovník

Dátový slovník bol vytvorený v programe Microsoft Excel 2010.

typ_Entity	typ_atribútu	dátový_typ	veľkosť	kľúč	NULL	INDEX	IO
rss_channels	rss_channels_id	smallint	2 Bytes	A	N	A	Auto Increment
	rss_channels_channel	varchar	150 Chars	N	N	N	
	rss_channels_server_url	varchar	100 Chars	N	N	N	
	rss_channels_selectors	varchar	100 Chars	N	N	N	
	rss_channels_last_read_article_ghash	int	4 Bytes	N	N	N	
articles	articles_guide_hash	int	4 Bytes	A	N	A	
	rss_channels_id	smallint	2 Bytes	N	N	A	foreign key from **rss_channels**
	articles_title	varchar	300 Chars	N	N	N	
	articles_link	varchar	400 Chars	N	N	N	
	articles_date	date	3 Bytes	N	N	N	
	articles_count	smallint	2 Bytes	N	N	N	
words	words_word	varchar	50 Chars	A	N	A	
	rss_channels_id	smallint	2 Bytes	A	N	A	foreign key from **rss_channels**
	words_date	date	3 Bytes	A	N	A	
	words_count	smallint	2 Bytes	N	N	N	
equivalent_words	equivalent_words_word	varchar	50 Chars	A	N	A	
	equivalent_words_root	varchar	50 Chars	N	N	N	
stop_words	stop_words_word	varchar	50 Chars	A	N	A	
rss_channels_categories	*rss_channels_id*	smallint	2 Bytes	A	N	A	foreign key from **rss_channels**
	categories_category	varchar	50 Chars	A	N	A	foreign key from **categories**
categories	categories_category	varchar	50 Chars	A	N	A	
users	users_name	varchar	50 Chars	A	N	A	
	users_password	varchar	50 Chars	N	N	N	
typ_Entity	typ_atribútu	dátový_typ	veľkosť	kľúč	NULL	INDEX	IO

Obrázok 14: Dátový slovník

B Ukážkové SQL dopyty

Dopyt pre výpis základných štatistík slov. Jedná sa o zobrazenie prvých 10 slov pre všetky servery a kategórie dohromady, pre každý deň zvlášť.

```
SELECT TOP 10 w.words_word, w.words_date, SUM(w.words_count) as TOTAL FROM words AS w
    GROUP BY w.words_word, w.words_date ORDER BY TOTAL DESC
```

Výpis 7: Ukážkový dopyt č.1

Dopyt pre zobrazenie prvých 10 štatistík slov pre kategóriu „šport“ za posledný týždeň.

```
SELECT TOP 10 w.words_word, SUM(w.words_count) as TOTAL FROM words AS w INNER JOIN
    rss_channels AS r ON w.rss_channels_id = r.rss_channels_id WHERE w.words_date BETWEEN
    '2012-06-22' AND '2012-06-29' GROUP BY w.words_word ORDER BY TOTAL DESC
```

Výpis 8: Ukážkový dopyt č.2

Dopyt pre zobrazenie prvých 20 slov pre server Idnes.cz, kategóriu „Ostrava“ a konkrétny deň.

```
SELECT TOP 20 w.words_word, SUM(w.words_count) as TOTAL FROM words AS w INNER JOIN
    rss_channels AS r ON w.rss_channels_id = r.rss_channels_id WHERE r.rss_channels_server_url=
    'http://www.idnes.cz/' AND w.words_date BETWEEN '2012-06-29' AND '2012-06-29'
    GROUP BY w.words_word ORDER BY TOTAL DESC
```

Výpis 9: Ukážkový dopyt č.3

Dopyt pre zobrazenie prvých 10 najdôležitejších článkov pre server Lidovky.cz, kategóriu „invisible dog“ a špecifický časový interval.

```
SELECT TOP 10 w.articles_title, w.articles_date, w. articles_link , w.articles_count as TOTAL
    FROM articles AS w INNER JOIN rss_channels AS r ON w.rss_channels_id = r.rss_channels_id
    WHERE r.rss_channels_server_url='http://www.lidovky.cz/' AND w.articles_date BETWEEN '
    2012-06-01' AND '2012-06-15' ORDER BY TOTAL DESC
```

Výpis 10: Ukážkový dopyt č.4

C Popis webovej aplikácie

Webová aplikácia bola navrhnutá a vytvorená s cieľom poskytovať čo najjednoduchšie ovládanie, aby užívateľ intuitívne vedel ako má s aplikáciou narábať a čo môže pri voľbe jednotlivých možností od systému očakávať.
Zároveň sa podarilo vytvoriť príjemné užívateľské rozhranie, ktoré komunikuje s užívateľom v anglickom jazyku, preto budeme pri popise systému uvádzať originálne názvy jednotlivých položiek. Popis najdôležitejších častí systému bude sprevádzaný obrázkami.

Ukážku stránky so štatistikami vo verejnom režime môžme vidieť na obrázku 15. V hlavičke webovej stránky sa nachádza pruh s pomenovaním aplikácie, názvom a logom katedry. Pod ním sa nachádza horizontálne menu. Na pravo od menu je umiestnený jednoduchý stavový panel zobrazujúci aktuálny stav aplikácie (*„running“* alebo *„stopped“*) a takisto informáciu o administrátorskom prihlásení. Súčasťou stavového panelu je aj možnosť prihlásenia do administrátorského režimu. Ďalej nasleduje telo webovej stránky, kde sa zobrazujú výsledky jednotlivých volieb. V päte sú uvedené informácie o autorovi.

Verejný režim

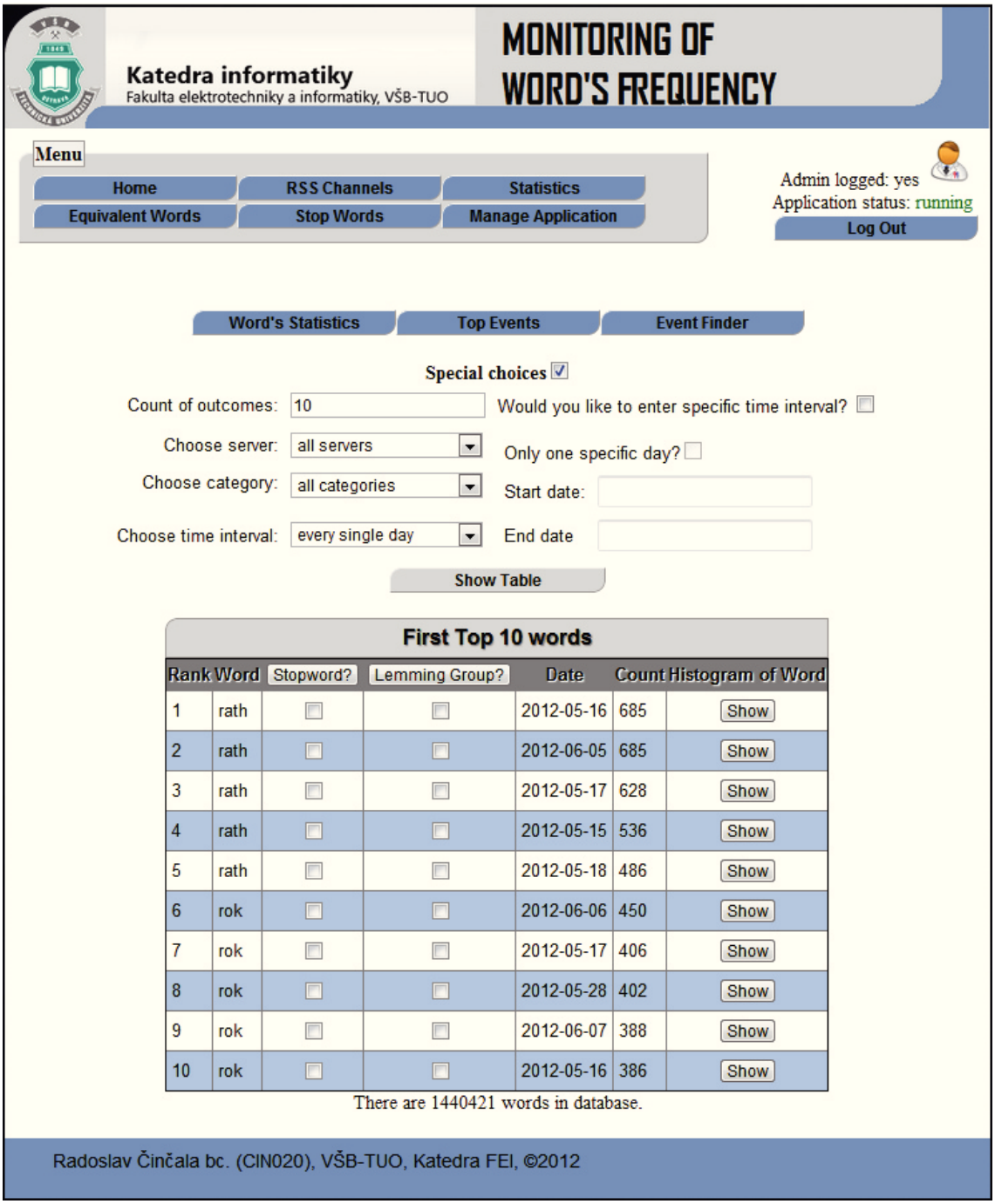
Na obrázku 15 vidieť v menu aplikácie, že vo verejnom režime má užívateľ okrem zobrazenia domovskej stránky, možnosť prezerať si zoznam monitorovaných RSS kanálov (v menu položka *„Rss Channels“*) a zobrazovať si štatistiky (položka *„Statistics“*). Domovská stránka ponúka stručný popis aplikácie. V zozname RSS kanálov je ku každému kanálu uvedená jeho URL adresa a príslušné selektory. Po umiestení kurzora na URL adresu kanálu, sa dodatočne zobrazí názov rodičovského serveru a kategórie kanálu, aby bola tabuľka prehľadnejšia.

Zobrazenie štatistík

Po zvolení štatistík máme na výber z troch ďalších možností. Sú to štatistiky slov (voľba *„Word's Statistics“*), štatistiky resp. prehľad článkov (voľba *„Top Events“*) a nakoniec vyhľadávanie v článkoch (voľba *„Event Finder“*). Články sú prezentované ako udalosti, ide iba o iný spôsob pomenovania.

Explicitne sú všetky štatistiky, či už štatistiky slov alebo článkov, zobrazené pre všetky servery a všetky kategórie dohromady, časovo pre každý deň zvlášť a v počte 10 zobrazených výsledkov.

Po zvolení voľby *„Special choices“*, môže užívateľ špecifikovať výsledky podľa konkrétneho serveru, kategórie a časového intervalu, prípadne obmedziť množstvo zobrazených výsledkov. To platí nie len pre slová, ale aj pre články. Pri výbere konkrétneho serveru sa vo výberovom polí pre kategórie zobrazia len tie, ktoré tento server ponúka.



Obrázok 15: Webová aplikácia: Zobrazenie štatistík (verejný režim)

Počet zobrazených výsledkov (*„Count of outcomes“*) sa musí pohybovať v intervale od 1 do 1000. Pri umiestení kurzora na formulárové pole o tom informuje dodatočná pripomienka, aby si bol užívateľ tohto obmedzenia vedomý. Iné hodnoty nie sú prípustné. Samozrejmosťou je povinnosť hodnoty v tomto poli a takisto typová kontrola.

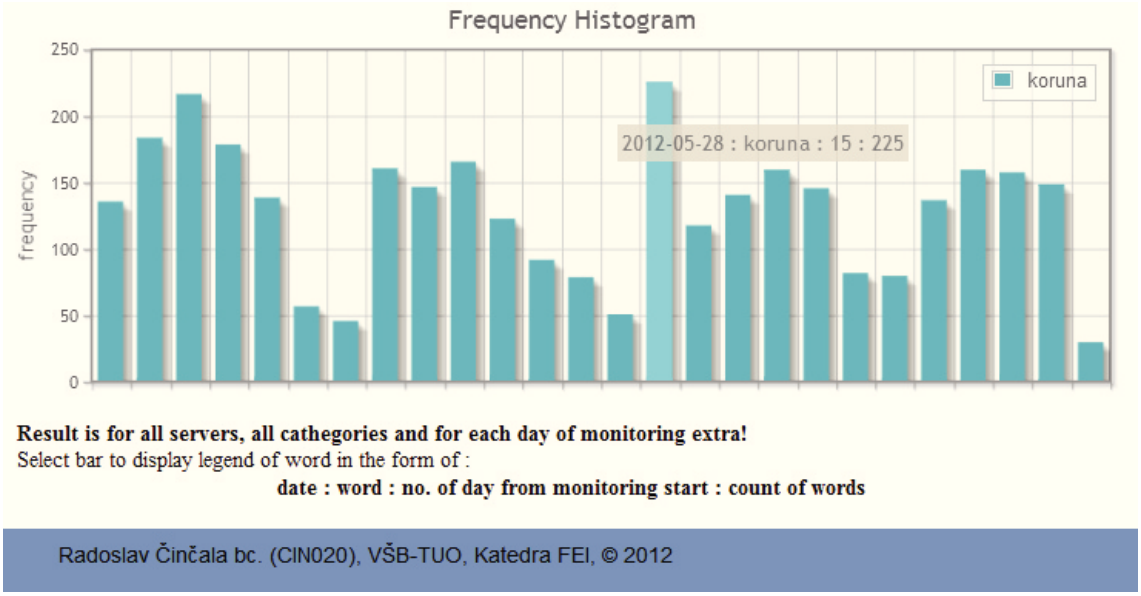
Časový interval ponúka okrem pôvodnej možnosti *„every single day“* aj možnosti *„Last week“*, *„Last month“*, *„Last year“* alebo *„Specific time interval“*, ktorý má možnosť si užívateľ zvoliť pomocou grafických kalendárov. Kalendáre umožňujú vybrať len dni od prvotného spustenia monitorovania vždy až do aktuálneho dňa (dni pre ktoré máme dáta). Pri využití tejto možnosti je povinná prítomnosť oboch dátumov a samozrejme musí platiť, že koncový dátum musí časovo nasledovať za počiatočným. Z toho vyplýva, že dátumy sa môžu rovnať, čím môžeme získať štatistiky pre konkrétny deň. Ak chceme docieliť štatistiky pre jeden deň, jednoduchšie je využiť možnosť *„Only one specific day“*, aby sme nemuseli zadávať dva rovnaké dátumy. Veľmi zaujímavé je použiť túto možnosť v rámci štatistík článkov. Pri zvolení konkrétnej kategórie môžme získať články z vybranej kategórie pre ľubovoľný deň.

Po zvolení parametrov si užívateľ zobrazí tabuľku pomocou voľby *„Show Table“*.

Čo sa týka štatistík slov, tabuľka obsahuje záznamy vo forme slovo, dátum a frekvencia. Dátum sa zobrazuje len vtedy, keď časový interval je nastavený na *„every single day“*, v opačnom prípade je dátumom zvolený časový interval. Súčasťou každého záznamu je možnosť zobrazenia grafických štatistík, o ktorých si povieme neskôr. Tabuľka je zotriedená zostupne podľa frekvencie slov.
Pri článkoch obsahuje tabuľka záznamy vo forme názov článku, dátum a *„dôležitosť“*. Názov článku slúži zároveň aj ako odkaz, po kliknutí na ktorý sa zobrazí článok v novej záložke prehliadača. Výsledky sú zotriedené podľa dôležitosti. Toto číslo vyjadruje koľko krát bol článok znovu zverejnený vo svojom rodičovskom RSS kanále z dôvodu jeho dôležitosti a opätovnej propagácie (viac bolo spomenuté na konci kapitoly 3.2). Pod oboma tabuľkami je uvedený aktuálny počet záznamov v databáze.

Veľmi zaujímavou možnosťou, ktorú môže užívateľ v rámci štatistík slov využiť je zobrazenie grafických štatistík pre vybrané slovo. Grafické štatistiky sú reprezentované histogramom, kde horizontálna os je pre dni a vertikálna znázorňuje frekvenciu slova. To umožňuje zistiť, kedy bolo slovná frekvencia v článkoch najvyššia. Na obrázku 16 sa nachádza histogram pre slovo „koruna“. Po umiestení kurzora na ľubovoľný stĺpec v grafe sa zobrazí popis, z ktorého môžeme na základe uvedenej legendy vyzistiť konkrétny deň a frekvenciu slova v tento deň.

Poslednou možnosťou je vyhľadávanie v článkoch (*„Event Finder“*). Táto funkcia ponúka jednoduchý formulár pre zadanie kľúčového slova v základnom tvare. Vyhľadávanie v článkoch prebieha tak, že po zadaní kľúčového slova, sú pomocou slovníka ekvivalentných slov vyhľadané jeho ekvivalentné formy. Tie sa následne vyhľadávajú v článkoch.



Obrázok 16: Webová aplikácia: Histogram pre slovo „koruna“

Výsledkom je zoznam nájdených článkov zotriedený podľa dôležitosti rovnako ako pri najdôležitejších článkoch. Jedná sa takisto o veľmi užitočnú funkcionalitu.

Administrátorsky režim

Po prihlásení do režimu administrácie pomocou voľby *„Login as Administrator“* s využitím príslušného mena a hesla sa užívateľovi zobrazia okrem možnosti verejného režimu ďalšie voľby umožňujúce správu aplikácie. O úspechu alebo neúspechu prihlasovania je užívateľ samozrejme informovaný. Ak prihlásenie prebehne úspešne, správcovi sa zobrazí stručný popis, ako pracovať s aplikáciou v tomto režime.

Správa RSS kanálov a pridávanie nových kanálov

V rámci položky *„RSS Channels“* má správca aplikácie možnosť upravovať zoznam monitorovaných RSS kanálov. Konkrétne sa jedna len o selektory kanálu, ostatné položky sú nemeniteľné alebo má možnosť kanál zmazať. Po zvolení tejto možnosti sa zobrazí varovné okno, ktoré informuje o tom, že so zmazaním kanálu budú zmazané všetky články a slová spájajúce sa s týmto kanálom. Správca svoje rozhodnutie buď potvrdí, alebo zruší.

Nad tabuľkou kanálov sa pri prihlásení do správcovského režimu automaticky objavuje nová voľba *„Add New Rss Channel“*, ktorá slúži na pridávanie nových RSS kanálov do monitorovania. Po zvolení tejto možnosti je užívateľ vyzvaný k zadaniu URL adresy kanála, ktorý chce pridať. Samozrejmosťou je povinnosť hodnoty v tomto formulárovom poli. Ďalej je požadovaná validná URL adresa. Aplikácia je takisto schopná určiť, či sa jedná

o adresu RSS kanálu a dokonca sa rozhoduje, či sa daný kanál už nachádza v zozname kanálov, alebo sa jedná o ešte nepriradený kanál. V prípad, že niektorá z vyše uvedených podmienok nutných pre pokračovanie nie je splnená, užívateľ je o tom okamžite informovaný.

V prípade úspechu sa dostávame k ďalšiemu kroku, ktorým je voľba metódy vyhľadávania selektorov. Metódy sú dve:

- Manuálna
- Automatická

Manuálna metóda vyžaduje znalosť teórie selektorov (kapitola 6.2), aby bol užívateľ schopný ručne nastaviť selektory pre RSS kanál. Súčasťou stránky umožňujúcej manuálne nastavenie selektorov je preto aj zdrojový kód ukážkového článku z tohto kanálu (obr. 17). Tu má užívateľ možnosť zobraziť si stránku v samostatnom okne prehliadača kliknutím priamo na názov článku, ktorý slúži zároveň ako odkaz. To je veľmi dôležité urobiť, pretože sa môže jednať o „netypický článok" obsahujúci napr. video, kde selektory nie je možné vyhľadať. V takom prípade volí možnosť *„Next Page"*, ktorej výsledkom je zmena ukážkového článku na v poradí ďalší článok z RSS kanálu. Po prejdení všetkých článkov sa začína zobrazovanie článkov od začiatku s tým, že v kanále môžu byť uverejnené nové články.
Po vyhľadaní množiny správnych selektorov je potrebné ich zadať do pripravených formulárov v spodnej časti stránky. Ak sú potrebné dva selektory, užívateľ volí možnosť *„Required"*, aby sa mu sprístupnilo nastavenie druhého selektoru. Samozrejmosťou je opäť povinnosť hodnôt v jednotlivých vstupných poliach. Ak si je tvorca selektorov istý svojím nastavením pokračuje možnosťou *„Build Selectors"*.

V nasledujúcom kroku sa zobrazia skonštruované selektory podľa príslušných pravidiel. Tu je potrebné priradiť ukladaný RSS kanál k rodičovskému serveru, ku ktorému sa viaže svojou príslušnosťou. Výber je možné uskutočniť z už „známych" serverov, alebo využiť možnosť vloženia nového serveru. Za názov serveru sa považuje jeho URL adresa, preto dochádza opäť ku kontrole správnosti zadaného URL. Už v tejto fáze môže správca svoje rozhodnutie o pridaní nového RSS kanálu zrušiť (voľba *„Discard"*), alebo prechádza na ďalší krok pomocou *„Next"*.

Posledným krokom manuálneho nastavovania selektorov je voľba kategórií RSS kanálu. Správca má možnosť vybrať si z už existujúcich kategórií kanálov, alebo zadať zoznam nových. Opäť je k dispozícii možnosť zrušenia celého procesu pridávania nového RSS kanálu. V opačnom prípade správca ukončuje svoje rozhodnutie voľbou *„Insert New RSS Channel"*. Kanál je následne pridaný do spracovania. V závislosti od stavu aplikácie, ak je aplikácia v stave spusteného monitorovania, sa automaticky spúšťa extrakcia článkov a slov z tohto kanálu. V opačnom prípade k prvotnému spracovaniu RSS kanálu dochádza až pri nasledovnom spustení sledovania.

Obrázok 17: Webová aplikácia: Manuálna metóda vyhľadávania selektorov

Manuálna metóda slúži ako doplnková metóda k metóde automatickej, ktorá nemusí byť vždy úspešná.

Automatická metóda spočíva v automatickom nastavení selektorov pre zvolený RSS kanál. Je to relatívne časovo náročná operácia, preto po zvolení tejto možnosti je užívateľ upozornený na časovú náročnosť tejto operácie v podobe informačného okna. Po potvrdení voľby sa spúšťa komplexný algoritmus, ktorého cieľom je nájsť selektory pre vybraný RSS kanál. O aktuálnom priebehu spracovania, je užívateľ informovaný ukazovateľom pomeru spracovaného k celkovému počtu článkov v RSS kanále.

Po skončená spracovania dostávame k dispozícií množinu nájdených a skonštruovaných selektorov spolu s percentuálnym vyjadrením ich „správnosti" vzhľadom na aktuálnu množinu článkov v RSS kanále (obr. 18). Číslo je percentuálnym vyjadrením počtu článkov z celkového počtu článkov, v ktorých boli vyhľadané výsledné selektory. Máme možnosť ich skontrolovať pomocou ukážkového zdrojového kódu podobne ako je tomu pri manuálnej metóde. Táto metóda nie je dokonalá, preto sa môže stať, že selektory neboli nájdené správne. V tom prípade je k dispozícií možnosť ich úpravy, opäť je však nutná teória k tvorbe selektorov.
Okrem toho v tejto časti volíme príslušnosť k rodičovskému serveru. Posledná úroveň je rovnaká ako pri manuálnej metóde - výber kategórií RSS kanálu. Nasledovný scenár je už známy.

Po pridaní nového informačného kanálu zabezpečí aplikácia automatické presmerovanie na zoznam monitorovaných RSS kanálov, kde si môže užívateľ skontrolovať novo pridaný kanál v zozname informačných kanálov.

Správa ekvivalentných slov

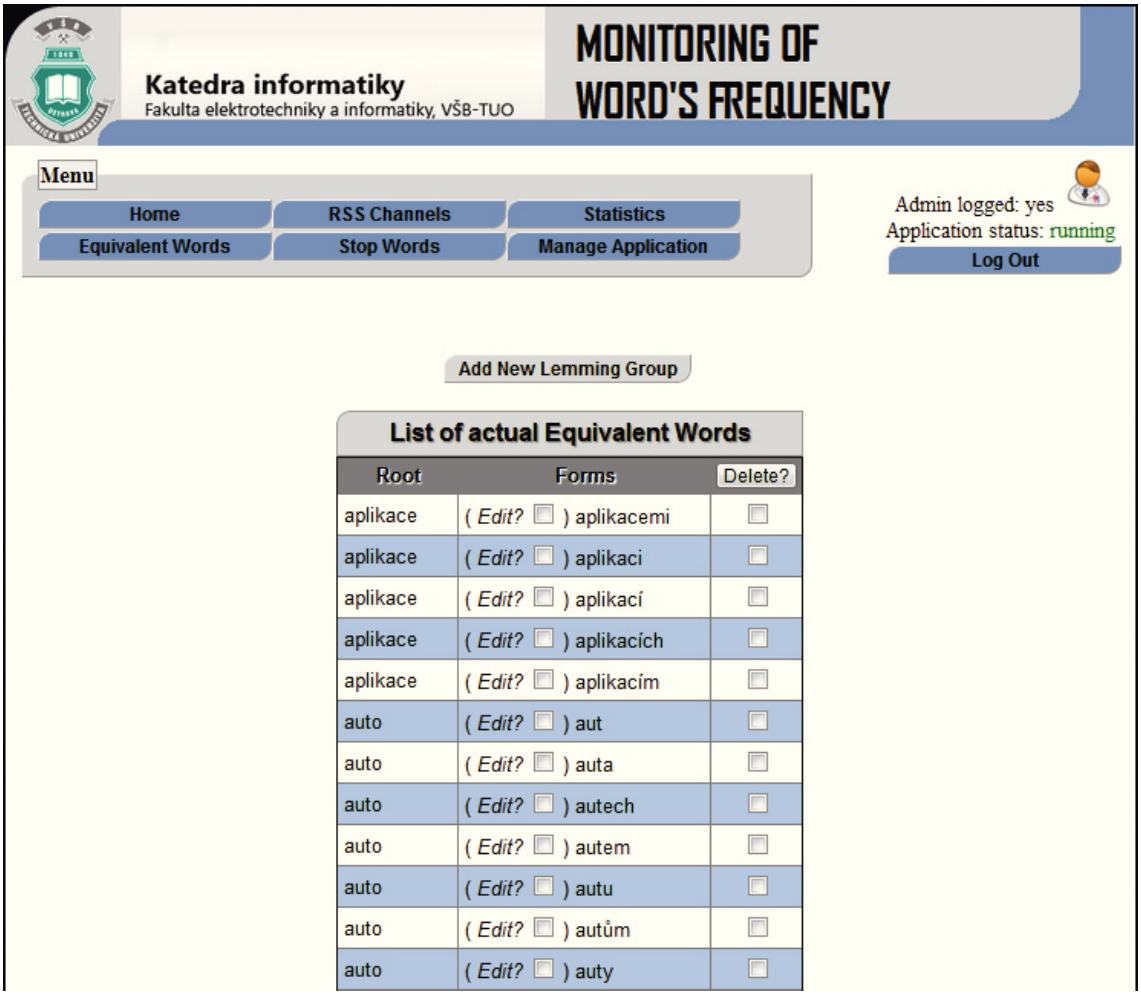
Všetko potrebné ohľadom ekvivalentných skupín slov bolo vysvetlené v kapitole 6.4.2. Pod menu položkou *„Equivalent words"* sa skrýva správa ekvivalentných slov, alebo tiež označovaných ako lematizačné skupiny. Správca má k dispozícií zoznam ekvivalentných slov zotriedený abecedne podľa koreňa slov. Priamo v tejto tabuľke je možné editovať formy slov alebo mazať jednotlivé záznamy (obr. 19). Vo vrchnej časti stránky a nachádza možnosť *„Add New Lemming Group"*, ktorá slúži na pridávanie nových lematizačných skupín. Skupiny sa zadávajú vo forme koreň slova a jednotlivé tvary oddelené čiarkou.

Samozrejmosťou je spätná lematizácia všetkých evidovaných slov podľa novej skupiny ekvivalentných slov.

Správa stop slov

Pod položkou *„Stop Words"* nájdeme správu stop slov. Potrebná teória ohľadom stop slov bola vysvetlená v kapitole 6.4.3. Opäť máme k dispozícií zoznam stop slov. Jednotlivé slová môžme priamo v tabuľke mazať. Editácia nie je potrebná, pretože záznamy sú re-

Obrázok 18: Webová aplikácia: Automatická metóda vyhľadávania selektorov



Obrázok 19: Webová aplikácia: Správa ekvivalentných slov

prezentované len jedným slovom, ktoré môžeme v prípade potreby zmazať a vložiť iné slovo. Na vkladanie nových slov slúži položka vo vrchnej časti stránky *„Add New Stop Word“*. Vkladáme zoznam slov oddelených čiarkou. Stop slová skladajúce sa z jedného písmena netreba ukladať do zoznamu stop slov, pretože sú zo spracovania automatický vyraďované. Samozrejmosťou je spätná aktualizácia databázy, kedy všetky slová označené ako stop slová budú vymazané.

Čo sa týka stop slov a ekvivalentných slov, má užívateľ s administrátorským oprávnením možnosť využiť vkladanie rovno zo zoznamu štatistík slov, kde je umožnené priamo označiť vybrané slová za stop slová alebo ako formy slova a uložiť ich tak do príslušného zoznamu.

Manažment aplikácie

Poslednou možnosťou v administrátorskom režime aplikácie je manažment aplikácie (voľba v menu *„Manage Application“*) (obr. 20). Správa aplikácie umožňuje najmä spustenie a zastavenie spracovania článkov (voľby *„Run Monitoring“* a *„Stop Monitoring“*).

Ďalšou dôležitou možnosťou je nastavenie časového intervalu (v minútach) pre opätovnú kontrolu RSS kanálov z dôvodu hľadania a spracovania nových článkov. Táto hodnota sa môže pohybovať v intervale od 10 do 60 minút. Iné hodnoty nie sú prípustné.
Časový interval musí byť nastavený tak, aby vyhovoval „najrýchlejšiemu“ RSS kanálu. Tým je myslený kanál, v ktorom sú články publikované najrýchlejšie. Dajme tomu, že v kanále je publikovaných vždy 25 najnovších článkov. Interval musí byť nastavený na takú hodnotu, aby za tento čas nebolo v RSS kanále publikované viac článkov ako je jeho veľkosť tj. viac ako 25 článkov. Hrozila by tak strata niektorých článkov. Toto konštatovanie vychádza z popisu fungovania RSS vo vzťahu k spravodajským serverom v kapitole 3.2. Hodnota okolo 30 minút je vhodná pre všetky kanály, preto sa nemusíme obávať straty článkov, avšak môže byť zmenená v dovolenom rozmedzí.

Poslednou možnosťou aplikačného manažmentu je sledovanie informačných logov, ktoré sa zobrazia po zvolení možnosti *„Information Logs“*. Jedná sa o všetky bežné záležitosti spojené s behom aplikácie. Napr. kedy bolo spustené resp. zastavené monitorovanie, informácie o administrátorskom prihlásení, nových stop slovách, lematizačných skupinách atď.

Obrázok 20: Webová aplikácia: Manažment aplikácie

D Obsah priloženého CD

Priložené CD obsahuje nasledujúce súbory a adresáre:

- `[javadoc]` - Programátorská príručka vygenerovaná pomocou nástroja JavaDoc.
- `[CIN020_NewsMonitoring]` - Zdrojové kódy (Eclipse Export).
- `[text]` - Elektronická podoba diplomovej práce vo formáte PDF.
- `[Prístup k Aplikácií]` - Popis prístupu k aplikácií.